NGA.STND.0063_1.1_WMTS

2018-04-27

NGA STANDARDIZATION DOCUMENT

National System for Geospatial-Intelligence (NSG) Web Map Tile Service 1.0.0 Interoperability Standard

(2018-04-27)

Version 1.1

NATIONAL CENTER FOR GEOSPATIAL INTELLIGENCE STANDARDS

Document type:	Profile of Standard
Document date:	27 April 2018
Edition number:	
Responsible Party:	Geospatial Web Services (GWS) Focus Group (FG)
Audience:	GWS participants and associates
Abstract:	
Copyright:	

Table of Contents

Table of Tables

Table of Figures

i. Submitting organizations

National Geospatial-Intelligence Agency (USA)

ii. Document Point of Contact (POC)

All questions regarding this document shall be directed to the editor or the contributors:

Person	Organization	Email
GWS FG Chair	National Center for Geospatial Intelligence Standards	NCGIS-mail@nga.mil

iii. Revision History

Date	Version Number	Date of Authorization to Publish	Change Description
9/15/2016	1.0	9/15/2016	Initial Release
2/23/2018	1.1		Technical and administrative edits including: 1) Clarification of Service Metadata Requirements 2) Clarification of NMF Metadata Requirements 3) Identification of two conformance classes: Basic and Queryable 4) Additions to Abstract Test Suite (ATS) 5) Development of an Implementation Conformance Statement (ICS) for server and client
4/27/2018	1.1		Adjudication of GWG Review comments

iv. Future Work

Two activities have been identified as potential future work items by the Geospatial Intelligence Standards Working Group (GWG) Geospatial Web Services Focus Group (GWS FG). These work items include: 1) development of a Reference Implementation for use by implementers and testers and 2) development of a client abstract test suite.

Foreword

The Web Map Tile Service (WMTS) described in this Interoperability Standard builds on earlier efforts to develop scalable, high performance services for web based distribution of cartographic maps. WMTS is inspired by the OSGeo[1]Tile Map Service Specification. The team that worked on this standard also considered similar initiatives, such as Google maps and NASA OnEarth. This OGC standard includes both resource (RESTful approach) and procedure oriented architectural styles, i.e., Key-Value Pair (KVP) and Simple Object Access Protocol (SOAP) encoding, in an effort to harmonize this interface standard with the OSGeo specification.

WMTS is based upon earlier efforts to develop services for the web based distribution of cartographic maps. The OGC WMTS provides a complementary approach to the OGC Web Map Service (WMS) for tiling maps. WMS provides a simple HTTP interface for requesting geo-registered map images from one or more distributed geospatial databases. A WMS request defines the geographic layer(s) and area of interest to be processed. The response to the request is one or more geo-registered map images (returned as JPEG, PNG, etc) that can be displayed in a browser application. The interface also supports the ability to specify whether the returned images should be transparent so that layers from multiple servers can be combined or not.

WMTS trades the flexibility of custom map rendering for the scalability possible by serving of static data (base maps) where the bounding box and scales have been constrained to discrete tiles. The fixed set of tiles allows for the implementation of a WMTS service using a web server that simply returns existing files stored on the server. The fixed set of tiles also enables the use of standard network mechanisms for scalability such as distributed cache systems (Normative Reference [1]).

NSG profiles seek to maximize commonality and improve interoperability across different standards. To this end, the coordinate reference systems and projections are the same for the NSG Profiles of WMS and WMTS. In addition, the coordinate reference systems, projections, and tiling schemes are the same for the NSG Profiles of WMTS and GeoPackage.

Executive Summary

This Interoperability Standard, also known as a profile, identifies and tailors the implementable provisions prescribed for the National System for Geospatial-Intelligence (NSG) for a Web Map Tile Service (WMTS) based on the Open Geospatial Consortium (OGC) OpenGIS® Web Map Tile Service Implementation Standard; Reference Number OGC 07-057r7, version 1.0.0 dated 2010-04-06. It is a Class 2 conformance profile as defined by ISO 19106, Geographic Information – Profiles date 2004-07-01 in that it includes a single standard, in this case OGC 07-057r7, with permitted NSG extensions and/or restrictions to the standard. An Interoperability Standard provides detailed direction on how to use the clauses, options, and parameters of the base standard(s). This guidance

[1] OSGeo is an independent non-profit legal entity established to support needs of the open source geospatial community.

is designed to be specific enough for any two independent and compliant software implementations to ‘plug and play’ with each other.

The framework, concepts, and methodology for testing, and the criteria to be achieved to claim conformance are specified in the OGC Compliance Testing Policies and Procedures and the OGC Compliance Testing web site[2].

To ensure the ability of implementing this profile, existing constraints among submitting organizations and vendors have been taken into account, in order to distinguish between normative specifications, recommendations and future work directions. Normative specifications include both extensions/restrictions of WMTS and system requirements specifications in order to enable interoperability by appropriate configuration of existing software. System requirements specifications are intended to be applied in the design and fielding of systems having services compliant to this Interoperability Standard. They come along with a rationale and conformance tests which provide guidelines for testing compliance of implementations.

The WMTS interface offers 3 Operations:

1. GetCapabilities (mandatory): Allows for obtaining Service Metadata
2. GetTile (mandatory): Returns a map tile
3. GetFeatureInfo (optional): Provides additional information at a pixel location in the pictures of maps that were returned by previous GetTile requests

1. Introduction

The OGC WMTS Implementation Standard provides an interface to serve digital maps using predefined image tiles. It complements the existing OGC Web Map Service (WMS) standard. As described in Normative Reference [1], “The WMS standard focuses on flexibility in the client request enabling clients to obtain exactly the final map image they want. A WMS client can request that the server creates a map by overlaying an arbitrary number of the map layers offered by the server, over an arbitrary geographic bound, with an arbitrary background color at an arbitrary scale, in any supported coordinate reference system. The client may also request that the map layers be rendered using a specific server advertised style or even use a style provided by the client when the WMS server implements the OGC Styled Layers Descriptor (SLD) standard. However, all this flexibility comes at a price: server image processing must scale with the number of connected clients and there is only limited potential to cache images between the server and client since most images are different”.

The tile resource is generally a rectangular image containing 2D map data. Alternatively, this resource might be a non-image representation of the tile such as a description of the tile or a link to

[2] http://cite.opengeospatial.org/teamengine/

the actual image. For example, the tile resource could be a KML document used in a super-overlay, or a tile metadata document. When returning an image tile, a full single tile will always be returned. Also, the background pixels of a tile should be transparent when possible so that the client can overlay the tiles on top of other map data (possibly other tiles).

To improve performance, instead of creating a new image for each request, a WMTS returns small pre-generated images (e.g., PNG or JPEG) or reuses identical previous requests that follow a set of tile matrices. This standard provides support for multiple architectural patterns - KVP, REST and SOAP. WMTS is the first OGC standard to include a RESTful approach in addition to the usual OGC encodings.   Additionally, WMTS provides a natural way to evolve WMS services into a more constrained but more scalable and faster service.

2. Scope

This document defines specific National System for Geospatial-Intelligence (NSG) requirements, recommendations, and guidelines for implementation of the Open Geospatial Consortium (OGC) OpenGIS® Web Map Tile Service Implementation Standard, Version 1.0.0, OGC 07-057r7.

3. Conformance

The NSG Interoperability Standard establishes two conformance classes:

1. NSG Basic WMTS (includes GetCapability and GetTile)
2. NSG Queryable WMTS (extends NSG Basic WMTS by adding GetFeatureInfo)

Conformance with this NSG WMTS Interoperability Standard shall be checked using all relevant tests specified in Annex A (normative) of this document. In addition, conformance shall be checked using the framework, concepts, and methodology for testing specified in the OGC Compliance Testing Policies and Procedures and the OGC Compliance Testing web site[3]. Normative server requirements for Basic and Queryable are summarized in Table 1 below and described in more detail as follows: Tables 4 and 5 for Basic, and Table 16 for Queryable. Client requirements are described in Table 18. Abstract tests for checking server implementations of this standard are located in Annex A. Client abstract tests are identified as a future work item in Section iv.

Table 1: NSG WMTS Profile Conformance Classes

Conformance class name	**Operation or behavior**	**NSG WMTS Conformance Test**
NSG Basic WMTS	**NSG Basic WMTS (normative)** Requirement 1 to Requirement 16	A.2 Server test Module (NSG Basic WMTS)
NSG Queryable WMTS	**NSG Queryable WMTS (normative)** All requirements from NSG Basic WMTS and Requirement 17 to Requirement 22	A.3 Server test Module (NSG Queryable WMTS)

4. References

[3] http://cite.opengeospatial.org/teamengine/

Normative references in the NSG WMTS Interoperability Standard are identified in Table 2.

Table 2: Normative References in NSG WMTS Interoperability Standard

ID	Standard or Specification	Description of Service
1	OpenGIS® Web Map Tile Service Implementation Standard, Version: 1.0.0, OGC 07-057r7, 2010-04-06	A WMTS enabled server application can serve map tiles of spatially referenced data using tile images with predefined content, extent, and resolution.
2	OGC Web Service Common Implementation Specification, Version 2.0.0, OGC 06-121r9, 2010-04-07	Specifies many of the aspects that are, or should be, common to all or multiple OGC Web Service (OWS) interface Implementation Standards.
3	National System for Geospatial Intelligence (NSG) Metadata Foundation (NMF), Version 3.0, 31 August 2016, NGA.STND.0012_3.0	Defines the minimum mandatory geospatial metadata for datasets, series, and services in the NSG. The NMF 3.0 profile contains the minimum mandatory core set of geospatial metadata elements, agreed upon via international consensus and extended via DoD and IC requirements.
4	NSG Application Schema (NAS) - Part 1: Platform Independent Model, Version 8.0, 28 Jul 2016, NGA.STND.0022_8.0_NAS	Specifies the Platform Independent Model (in accordance with the Object Management Group (OMG) Model-Driven Architecture (MDA)) that determines the syntactic structure used across the NSG to represent the geospatial semantics specified by NSG Core Vocabulary (NCV), NSG Taxonomy (NTAX), and NSG Enterprise Ontology (NEO).
5	Implementation Practice Web Mercator Map Projection, NGA.SIG.0011_1.0.0_WEBMERC 1.0.0, 2014-02-18	Contains general technical information on the Web Mercator map projection to include its relationship to the World Geodetic System 1984 (WGS 84) and the risks and impacts associated with its usage as a positioning reference tool for the NSG.

ID	Standard or Specification	Description of Service
6	Geography Markup Language (GML) simple features profile (with Corrigendum), Version: 2.0, OGC 10-100r3, 2012-04-05	GML Simple Features Profile that specifies restricted subset of simple geometry types.
7	W3C SOAP Version 1.2 Part 1: Messaging Framework, W3C Recommendation, 2007-04-27	A lightweight protocol intended for exchanging structured information in a decentralized, distributed environment.
8	W3C SOAP 1.2 Attachment Feature, W3C Working Group Note, 2004-06-08	Defines a SOAP feature that represents an abstract model for SOAP attachments. Provides the basis for creation of SOAP bindings that transmit such attachments along with a SOAP envelope, and provides for reference of those attachments from the envelope.
9	XML Data Encoding Specification for Information Security Markings, Version 13, 2014-09-05	Information Security Marking (ODNI standard)

4.1. Informative References

The documents listed in Table 3 contain useful information to augment NSG understanding and application of the material in this interoperability standard in conjunction with the actual standard profiled.

Table 3: Informative References in NSG WMTS Interoperability Standard

Reference ID	Standards and Specifications Title
ISO 19106:2004(E), 2004-07	Geographic information — Profiles
IETF RFC 2119, S. Bradner, Harvard University, 1997-03-14	Key Words for use in RFCs to Indicate Requirement Levels
OGC 03-040, v0.1.3, 2003-09-16	OpenGIS® Reference Model
OGC 06-004r3, v1.0.0, 2006-02-28	Geospatial Digital Rights Management Reference Model (GeoDRM RM)

Reference ID	Standards and Specifications Title
NGA.STND.0065_1.0_NTAX, 2017-04-12	National System for Geospatial Intelligence Taxonomy (NTAX) Standard Edition 1.0, 12 April 2017
NGA.STND.0066_1.0_NCV, 2017-06-01	National System for Geospatial Intelligence (NSG) Core Vocabulary (NCV) Standard (2017-06-01) Edition 1.0
NGA.STND.0064_1.0, 2017-08-15	National System for Geospatial Intelligence Enterprise Ontology (NEO) Standard, Edition 1.0
NAS Part 2 v8.0 XSD, 2016-07-18	NAS - Part 2: Platform Specific Model - XML Schema, Version 8.0
OGC 13-082r2, 2016-01-19	OGC® Web Map Tile Service (WMTS) Simple Profile

5. Terms, Definitions, and Abbreviations

Terms and Definitions

For the purposes of this document, terms and definitions found in **ISO 19142:2010** / **[OGC 09-025r1]** apply.

Abbreviations

ASP	Active Server Pages
CGI	Common Gateway Interface
CRS	Coordinate Reference System
FG	Focus Group
GML	Geography Markup Language
GWS	Geospatial Web Services
HTTP	HyperText Markup Language
JPEG	Joint Photographic Experts Group
ISO	International Organization for Standardization
KVP	Key-Value Pair
NAS	NSG Application Schema
NCV	NSG Core Vocabulary
NEO	NSG Enterprise Ontology
NGA	National Geospatial-Intelligence Agency

NSG	National System for Geospatial-Intelligence
NMF	NSG Metadata Foundation
NTAX	NSG Taxonomy
OGC	Open Geospatial Consortium
OWS	OGC Web Services
PNG	Portable Network Graphics (Image Format)
REST	Representational State Transfer
SLD	Styled Layer Descriptor
SOAP	Simple Object Access Protocol
UML	Unified Modelling Language
URL	Uniform Resource Locator
WGS 84	World Geodetic System 1984
WMS	Web Map Service
WMTS	Web Map Tile Service
XML	Extensible Markup Language

6. Introduction

A WMTS that complies with the NSG WMTS Interoperability Standard **shall**

1. satisfy all requirements stipulated in the OGC WMTS 1.0.0 Standard;
2. satisfy all requirements stipulated in this document.

This profile provides advice on implementation of the WMTS so that tests can be provided to ensure objective compliance to the profile. The profile provides a “Normative Clause” to describe how each component shall be implemented. The Normative Clause defines requirements where mandatory compliance is required for attainment of conformance. However, the profile also includes optional recommendations that may require a subjective test.

The following syntax is used to indicate the compliance requirement within the profile:

- Mandatory (M) – The requirement shall be implemented
- Conditional (C) – Mandatory when “If” statement applies
- Optional (O) – Should be implemented

7. NSG Basic WMTS

7.1. Normative Requirements

The Normative requirements (mandatory) for a WMTS 1.0.0 Basic server implementation as required by this Interoperability Standard are summarized in Table 4. Numbering is sequential and linked to the specific Requirement number as defined within this document.

Table 4: NSG Basic WMTS Normative Server Requirements

No.	Category	Modifier	NSG Requirement	Compliance	Section
1	Service Type	N/A	An NSG server shall be compliant to the NSG Basic WMTS conformance class.	M	7.4
2	Architectural Patterns/Encoding	N/A	An NSG WMTS server shall support REST and KVP.	M	7.5
3	GetCapabilities	KVP	An NSG WMTS server shall declare its support for GetCapabilities operations using KVP with HTTP GET by providing an OperationsMetadata section in the ServiceMetadata document with an Operation section for each supported HTTP request type.	M	7.6
4		SOAP	An NSG WMTS server shall generate a ServiceMetadata document, in response to a SOAP encoded GetCapabilities request, that looks like the one in Normative Reference 1, section 7.1.1.3, wrapped in a SOAP 1.2 envelope.	O	7.6
5		REST	An NSG WMTS server shall generate a ServiceMetadata document in response to a GetResourceRepresentation request in REST architecture that looks like the one described in section 7.1.1.3 of Normative Reference 1.	M	7.6
6		KVP	An NSG WMTS server shall respond to a GetTile operation request with a tile map that complies with the requested parameters.	M	7.7
7		SOAP	An NSG WMTS server shall	O	7.7

No.	Category	Modifier	NSG Requirement	Compliance	Section
	GetTile		respond to a SOAP-encoded GetTile operation request with an image in the MIME type specified by the Format parameter of the request, wrapped in the SOAP version 1.2.		
8		REST	An NSG WMTS server shall provide standard endpoints from which a representation of each Tile resource can be obtained.	M	7.7
9	Coordinate Reference Systems	N/A	An NSG WMTS server shall support the World Geodetic System WGS 84 expressed as: ▪ Geographic latitude, then longitude, recorded in decimal degrees - “EPSG::4326” (preferred) or ▪ Geographic longitude, then latitude recorded in decimal degrees - “CRS:84”	M	7.8
10	Projections	N/A	An NSG WMTS server shall support the following projections whose validity zones overlap data published by the service: • World Mercator Projection - “EPSG::3395” • UPS projection over WGS84 (north zone) - “EPSG::5041” • UPS projection over WGS84 (south zone) - “EPSG::5042”	M	7.9
11	Well-Known Scale Sets	N/A	An NSG WMTS server shall employ the Well-Known Scale Sets identified in Annex C. NSG WMTS tiles conform to the Annex C WKSS, which are derived from the OGC WMTS Simple profile and direct implementation of tiles for mandated CRS and projections (see Rqmt 9 and 10 respectively) in 256 x 256 pixel tiles.	M	7.10
12	Tile Formats	N/A	An NSG WMTS server shall be	M	7.11

No.	Category	Modifier	NSG Requirement	Compliance	Section
			capable of offering tiles in the image/png, image/jpeg, and image/gif file formats.		
13	Negotiation of Standard Version	KVP	An NSG WMTS server shall support negotiation of the standard version used for client-server interactions.	M	7.12
14		SOAP	An NSG WMTS server shall support negotiation of the standard version used for client-server interactions.	O	7.12
15	Key Word List	N/A	An NSG WMTS server shall provide a keyword list based upon the NSG Metadata Foundation (NMF).	M	7.13
16	Cacheable Resources	N/A	An NSG WMTS server shall provide caching information (expiration date) for the data.	M	7.14

7.2. Non-Normative Requirements

Non-normative requirements for a WMTS 1.0.0 Basic server implementation as required by this Interoperability Standard are summarized in Table 5.

Table 5: Non-Normative Requirements

No.	Category	Modifier	Recommendation	Compliance	Section
1	MIME Type	N/A	If a service requires feature coordinate information, it is recommended that FeatureInfo documents be offered in the MIME type format "application/gml+xml; version=3.2".	O	7.15
2	Order of Variables and Values in URL Template	N/A	Any possible order of the variables and values in the URL template is valid. Nevertheless, recommend the following order: style, firstDimension, ..., lastDimension, TileMatrixSet, TileMatrix, TileRow, TileCol, J and I.	O	7.15
3	Themes	N/A	It is recommended that the Themes section of a WMTS service metadata document contain data about how layers are organized thematically.	O	7.15
4	Transparency	N/A	Clients should have the ability to support transparency.	O	7.15

7.3. Service Structure

The goal for implementing a WMTS enabled service is performance and scalability. Therefore, servers must be able to return tiles quickly. A good way to achieve that is to use locally stored, pre-rendered tiles that will not require any image manipulation or additional geo-processing. Server developers will decide if pre-rendered tiles will be generated in a previous tile-preparation process or generated on the fly utilizing a caching mechanism. With tile-based mapping it is important that the server is able to handle asynchronous access to tiles as most clients will simultaneously query for multiple tiles to fill a single view. The purpose of a WMTS service is to serve maps divided in individual tiles.

The WMTS interface allows a client to receive three types of resources either in response to a resource request in the resource oriented architectural style or in response to an operation in the procedure oriented architectural style. Those resources and operations are:

a) A **ServiceMetadata** resource (in response to a GetCapabilities operation for the procedure oriented architectural style) (required implementation by servers) – This resource describes the abilities and information holdings of the specific server implementation. In procedure oriented architectural style this operation also supports negotiation of the standard version being used for client-server interactions.

b) A **tile** resource (in response to a GetTile operation for the procedure oriented architectural style) (required implementation by servers) – This resource shows a fragment of a map representation of a layer.

c) A **FeatureInfo** resource (in response to a GetFeatureInfo operation for the procedure oriented architectural style) – This resource provides information about the features located at a particular pixel of a tile map, in a similar way to the WMS GetFeatureInfo operation, by providing, for example, the thematic attribute name and value pairs in textual form.

The RESTful pattern provides the ability to set up conformant WMTS servers simply. If all the tile images are pre-rendered, a WMTS server could even be created using no image processing logic at all but relying only on a normal web server to return the static ServiceMetadata XML document and provide the image tile files. This is important for deployment purposes as many Internet service providers (especially the free ones) allow web pages and static content hosting but do not, for security reasons, allow using Common Gateway Interface (CGI), Active Server Pages (ASP), or more advanced applications. The RESTful approach, therefore, enables small organizations to provide geographic data using readily available services or simple web server configurations. This approach also scales dramatically since the issues of serving fixed resources in high volumes have been continuously tackled over the past decades. Finally, this approach can benefit from network scaling effects since the images are considered by the HTTP protocol to be standard web resources and network providers can leverage their existing technologies to improve the flow of those resources to requesting clients.

These operations have many similarities to other OGC Web Services (OWS), including the Web Map Service (WMS), Web Feature Service (WFS), and Web Coverage Service (WCS). Many of the aspects of this WMTS interface that are shared in common with other OWSs are specified in the OpenGIS® Web Services Common Implementation Specification [OGC 06-121r3]. Many of these common aspects are included normatively by reference to that document, instead of being repeated.

The WMTS serves a single tile of a single layer of a map. Unlike WMS, there is no specified way to request a server to combine and return a map tile with information coming from more than one layer in a single fetching process. WMTS clients that want to show a combination of layers must make independent requests for the layer tiles and then combine or overlay the responses. Also bounding boxes and scales of these WMTS tiles are constrained to a discrete set of values (Normative Reference 1).

7.4. Service Type

According to this NSG WMTS Interoperability Standard, two conformance classes are defined: "Basic WMTS" and "Queryable WMTS". A Basic WMTS supports the GetCapabilities and GetTile operations (requests and responses); A Queryable WMTS includes all Basic WMTS operations and an additional GetFeatureInfo operation.

Requirement 1: A Basic NSG WMTS server shall be compliant with the Basic WMTS conformance class.

According to this requirement, an NSG WMTS server shall support the GetCapabilities and GetTile operations. This requirement ensures compatibility with the base standard.

7.5. Architectural Styles

The OGC WMTS Implementation Standard defines a standardized approach to declaring the images which a client can request from a server, enabling a single type of client to be developed for all servers. The standard specifies two different architectural styles, a procedure oriented architectural style and a resource oriented architectural style. For the former architectural style, there are several exchange mechanisms between clients and servers identified, including messages encoded using KVP, XML messages, or XML messages embedded in SOAP envelopes. The standard also defines the request mechanisms and endpoint publishing strategy to enable a resource oriented architectural style based on web based URL endpoints allowing clients to simply request resources as documents. NSG WMTS clients and servers are encouraged to support as many interfaces, i.e., KVP, RESTful, and SOAP, as possible to improve interoperability.

NSG Requirement 2: An NSG WMTS server shall support REST and KVP architectural patterns/encoding techniques.

An NSG WMTS server shall support the resource oriented architectural style with REST encodings and procedural oriented architectural style with KVP.

NSG NOTE 1: The intent of NSG Requirement 2 is to increase interoperability opportunities with NSG WMTS servers. However, based upon the operational requirements for a specific system, a program may elect to reduce the number of supported architectural patterns/interfaces. It is anticipated that increased security concerns regarding network vulnerability may drive implementations to adopt stronger security measures especially those enabled by SOAP encodings.

7.6. GetCapabilities Requests

NSG Requirement 3: An NSG WMTS server shall declare its support for GetCapabilities operations using KVP with HTTP GET.

This requirement is supported by providing an OperationsMetadata section (see Table 6) in the ServiceMetadata document with an Operation section for each supported HTTP request type.

Table 6: OperationsMetadata Parameters

<Element name	OGC Mandatory/ Optional	NSG Mandatory/ Optional	Mandated Value	Description
Operation	M	M		Metadata for one operation that this server interface implements. See following XML example for structure.
Parameter	O	O		Parameter valid domain that applies to one or more operations which this server implements.
Constraint	O	O		Constraint on valid domain of a nonparameter quantity that applies to this server.
Extended Capabilities	O	O		Metadata about server and software additional abilities

```
<ows:Operation name="GetCapabilities">
    <ows:DCP>
      <ows:HTTP>
        <ows:Get xlink:href="http://service.server/service/wmts?">
          <ows:Constraint name="GetEncoding">
            <ows:AllowedValues>
              <ows:Value>KVP</ows:Value>
            </ows:AllowedValues>
          </ows:Constraint>
        </ows:Get>
      </ows:HTTP>
    </ows:DCP>
  </ows:Operation>
  <ows:Operation name="GetTile">
    <ows:DCP>
      <ows:HTTP>
        <ows:Get xlink:href="http://service.server/service/wmts?">
          <ows:Constraint name="GetEncoding">
            <ows:AllowedValues>
              <ows:Value>KVP</ows:Value>
            </ows:AllowedValues>
          </ows:Constraint>
        </ows:Get>
      </ows:HTTP>
    </ows:DCP>
  </ows:Operation>
  <ows:Operation name="GetFeatureInfo">
    <ows:DCP>
      <ows:HTTP>
        <ows:Get xlink:href="http://service.server/service/wmts?">
          <ows:Constraint name="GetEncoding">
            <ows:AllowedValues>
              <ows:Value>KVP</ows:Value>
            </ows:AllowedValues>
          </ows:Constraint>
        </ows:Get>
      </ows:HTTP>
```

```
      </ows:DCP>
    </ows:Operation>
  </ows:OperationsMetadata>
```

NSG Requirement 4: An NSG WMTS server shall generate a ServiceMetadata document in response to a SOAP encoded GetCapabilities request.

The ServiceMetadata document shall conform to the example provided in Normative Reference 1, section 7.1.1.3, wrapped in a SOAP 1.2 envelope as described in Normative Reference 7.

NSG Requirement 5: An NSG WMTS server shall generate a ServiceMetadata document in response to a GetResourceRepresentation request in REST architecture.

In response to a valid request for a ServiceMetadata representation from a client, an NSG WMTS server will generate a document that populates all mandatory elements shown in Figure 1 and described in Tables 7 through 12 derived from OWS Common (Normative Reference 2). The following does not re-define the attributes in OWS Common, rather provides additional guidance on elements with NSG applications. Unless otherwise noted, full structure information should be referenced from OWS Common.

Figure 1: ServiceMetadata UML model

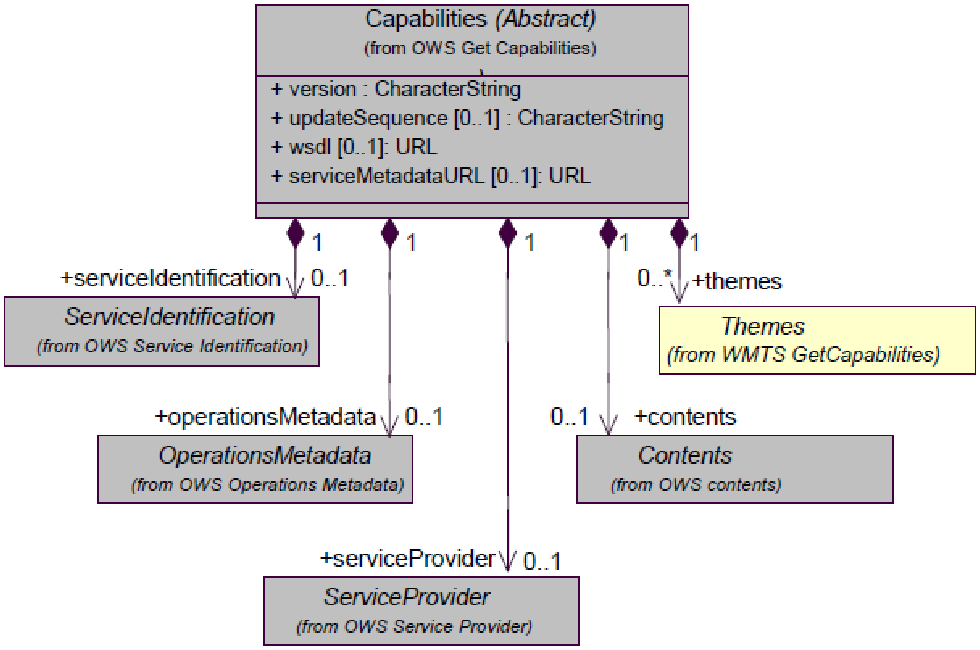

Capabilities (Abstract)
(from OWS Get Capabilities)
+ version : CharacterString
+ updateSequence [0..1] : CharacterString
+ wsdl [0..1]: URL
+ serviceMetadataURL [0..1]: URL
1
1
1
1
1
+serviceIdentification
0..1
ServiceIdentification
(from OWS Service Identification)
0..*
+themes
Themes
(from WMTS GetCapabilities)
+operationsMetadata
0..1
OperationsMetadata
(from OWS Operations Metadata)
0..1
+contents
Contents
(from OWS contents)
+serviceProvider
0..1
ServiceProvider
(from OWS Service Provider)

Table 7: Service Identification Parameters

Element name	OGC Mandatory/ Optional	NSG Mandatory/ Optional	Mandated Value	Description
Service Type	M	M	OGC WMTS	A service type URN from a registry of services, normally used for machine-to-machine communication. Shall state "OGC WMTS".
Service Type Version	M	M	1.0.0	Version of OGC Standard.
Profile	O	M	NSG 1.1	Identification of Interoperability Standard.
Title	M	M		Title of this server, normally used for display to a human.
Abstract	O	M	This service implements the NSG WMTS 1.0.0 InteroperabilityStandard Version 1.1, Basic conformance class.*	Brief narrative description of this server, normally available for display to a human. Must contain mandated value.
Keywords	O	M		See NSG Requirement 15.
Fees	O	O		Fees and terms for using this server.
Access Constraints	O	M		Shall be populated with the highest level of classification. The classification level shall be encoded using the most recent version of the IC Information Security Marking standard (IC.ISM) Normative Reference 9. The classification level encoded in the IC.ISM markings shall be sufficiently restricted to

Element name	OGC Mandatory/ Optional	NSG Mandatory/ Optional	Mandated Value	Description
				dominate any content which may reside on the service.

*If Queryable conformance class is implemented, see NSG Requirement 18 for appropriate text.

Table 8: Service Provider Parameters

Element name	OGC Mandatory/ Optional	NSG Mandatory/ Optional	Mandated Value	Description
Provider Name	M	M		Unique identifier for service provider organization.
Provider Site	O	O		Reference to the most relevant web site of the service provider.
Service Contact	M	M		Information for contacting service provider. See Table 9 for structure.

Table 9: Service Contact Parameters

Element name	OGC Mandatory/ Optional	NSG Mandatory/ Optional	Mandated Value	Description
Individual Name	O	O		Name of the responsible person- surname, given name, title separated by a delimiter.
Organization Name	M	M		Name of Organization. See Table 49 National System for Geospatial Intelligence (NSG) Metadata Foundation (NMF), Version 3.0 (Normative Reference 3).
Position Name	O	O		Role or position of the responsible party. See Table 48 (Normative Reference 3).
Contact Info	O	O		See Table 45 (Normative Reference 3).
Role	O	O		See Table 46 (Normative Reference 3).

Table 10: Content Parameters

Element name	OGC Mandatory/ Optional	NSG Mandatory/ Optional	Mandated Value	Description
Layer	O	M		Metadata describing one top level dataset available from this server. One for each layer, see Table 11 for structure.
Other Source	O	O		Reference to another source of contents metadata.
Tile Matrix Set	O	M		A description of the geometry of a tile cut. See NSG Requirement 9, 10, 11.

Table 11: Layer Parameters

Element name	OGC Mandatory/ Optional	NSG Mandatory/ Optional	Mandated Value	Description
Identifier	O	M		An unambiguous reference to this layer, normally used by software.
Title	O	M		Title of this layer, normally used for display to a human.
Abstract	O	M		Brief narrative description of this layer, normally available for display to a human.
Keywords	O	M		See NSG Requirement 15.
WGS84 Bounding Box	O	O		Minimum bounding rectangle surrounding dataset, using WGS 84 CRS with decimal degrees and longitude before latitude. Sub clause 10.2 of OWS Common, Normative Reference 2.
Bounding Box	O	M		Minimum bounding rectangle surrounding the layer, in the supported CRS sub clause 10.2 of OWS Common, Normative Reference 2.
Style	M	M		See Table 12 for structure.
Format	M	M		See NSG Requirement 12.
InfoFormat	M	M		See Table 17 for guidance.

Element name	OGC Mandatory/ Optional	NSG Mandatory/ Optional	Mandated Value	Description
Dimensions	O	O		The number of dimensions in the referenced CRS (the length of a coordinate sequence)
Tile Matrix Set Link	O	O		
Resource URL	O	M		See NSG Requirement 21.

Table 12: Style Parameters

Element name	OGC Mandatory/ Optional	NSG Mandatory/ Optional	Mandated Value	Description
Identifier	O	M		An unambiguous reference to this layer, normally used by software.
Title	O	M		Title of this layer, normally used for display to a human.
Abstract	O	M		Brief narrative description of this layer, normally available for display to a human.
Keywords	O	M		See NSG Requirement 15. Keywords for specific layer only.
Legend URL	O	O		See Table 8 of Normative Reference 1 for structure.
Is Default	O	M		Boolean, the style that a client should select as the first choice (default style).

7.7. GetTile Requests

NSG Requirement 6: An NSG WMTS server shall respond to a GetTile operation request with a tile map that complies with the requested parameters.

NSG WMTS servers shall support KVP requests for representations of image tiles by declaring support for and correctly handling GetTile requests. The Tile resource representation shall be returned in the format specified in the request when the format has been advertised in the ServiceMetadata document as available for that tile resource.

NSG Requirement 7: An NSG WMTS server shall respond to a SOAP encoded GetTile operation request with an image in the MIME type specified by the Format parameter of the request.

“The response of a successful SOAP-encoded GetTile operation request SHALL be an image with the MIME type specified by the Format parameter of the request, wrapped in the SOAP version 1.2 envelope. If the image is binary (such as is the case with image/png and image/jpeg images), it SHALL be base64 encoded and placed within the following XML element:

```
<element name="BinaryPayload">
     <complexType>
          <sequence>
               <element name="Format" type="ows:MimeType"/>  <element
               name="PayloadContent" type="base64Binary"/>
          </sequence>
     </complexType>
</element>
```

The xs:base64Binary type and associated base64-encoding rules are defined in the XML Schema Part 2 W3C recommendation. MIME element includes the MIME type of the original BinaryPayload.

NSG NOTE 2: The reason for using embedded encoded data instead of linking to an external source is to allow secured implementations.

NSG Requirement 8: An NSG WMTS server shall provide standard endpoints from which a representation of each Tile resource can be obtained.

The ServiceMetadata document contains a list of Layer elements and each layer that is available to be retrieved shall have one or more <ResourceURL> elements with the “resourceType” attribute set to “tile” and a template attribute. The template attribute contains a URL template that can be converted to a URL by applying the rules shown in Normative Reference 1, Table 32, URL template variables and possible values for tile. A standard HTTP GET is used to request the tile in the format specified by the attribute “format” as shown in Normative Reference 1, Table 31, Parts of the URL Template data structure for tiles.

7.8. Coordinate Reference Systems

NSG Requirement 9: An NSG WMTS server shall support the World Geodetic System WGS 84 expressed as:

- ***Geographic latitude, then longitude, recorded in decimal degrees (preferred) - “EPSG::4326“ or***
- ***Geographic longitude, then latitude recorded in decimal degrees - “CRS:84“***

7.9. Projections

NSG Requirement 10: An NSG WMTS server shall support the following projections whose validity zones overlap data published by the service:

- ***World Mercator Projection - "EPSG::3395"***
- ***UPS projection over WGS84 (north zone) - "EPSG::5041"***
- ***UPS projection over WGS84 (south zone) - "EPSG::5042"***

7.10. Well-Known Scale Sets

NSG Requirement 11: An NSG WMTS server shall employ the Well-Known Scale Sets identified in Annex C. NSG WMTS tiles conform to the Annex C WKSS, which are derived from the OGC WMTS Simple profile and direct implementation of tiles for mandated CRS and projections (see Rqmt 9 and 10 respectively) in 256 x 256 pixel tiles.

The list of map projections and their tiling schemes for global coverage is given in Table 13.

.

Table 13: Tiling Schemes for Global Coverage

Tiled Projection Identifier	Map Projection	EPSG Code	Bounding Box	Zoom Levels	Description
MercatorFullyTiled	Mercator	3395	Square	0 to 24	Mercator between +/-85.05 degrees
WGS 84 (Unprojected)	N/A	4326	Equirectangular	0 to 23	(-90, -180) to (90, 180)
UPSNorthTiled	UPS North	5041	Square	0 to 24	UPS centered on North Pole
UPSSouthTiled	UPS South	5042	Square	0 to 24	UPS centered on South Pole

NSG NOTE 3: Web Mercator[4] is a de facto standard used for web mapping applications. While it is used by virtually all major online map providers, including Google Maps, Bing Maps, OpenStreetMap, Mapquest, Esri, and Mapbox, it is not equivalent to the World Mercator projection identified in Table 13. If using WMTS map data from a Volunteered Geographic Information (VGI) or commodity data source which is in the Web Mercator projection, it is highly recommended that your service warn users that this data is suitable for visualization use cases only.

[4] Definition of Well-known scale Set for Web Mercator is found in Normative Reference 1, Annex E, Table E.4-Definition of Well-known scale set GoogleMapsCompatible.

For a use case that requires precise locations and precise navigation (land, air, and sea) the World Mercator projection is mandated.

7.11. File Formats

NSG Requirement 12: An NSG WMTS server shall be capable of offering tiles in the image/png, image/jpeg, and image/gif file formats.

Formats vary in utility to support thematic mapping, image quality, and transparency. Servers shall provide .png, .jpeg, and .gif to support varying utility and requirements. The png image format is recommended where transparency is needed. The jpeg image format is recommended where image quality is needed and transparency is not. Since a GetTile operation can serve only one tile at a time, clients should have the ability to support transparency and also be able to overlap tiles from the same geographical area.

NOTE: Users should be aware that only .gif and .png support transparency.

7.12. Negotiation of Standard Version

NSG Requirement 13: An NSG WMTS server shall support negotiation of the standard version used for client-server interactions (KVP encoding).

NSG Requirement 14: An NSG WMTS server shall support negotiation of the standard version used for client-server interactions (SOAP encoding).

The GetCapabilities operation includes a version-negotiation mechanism, allowing the client and server to agree on a standard version on which to base all future communication. The information in Table 14 below is an excerpt from Normative Reference 1, Table 17, which describes the "accept Versions" parameter in the GetCapabilities operation request.

Table 14: Accept Versions Parameter in GetCapabilities Operation Request

Names	Definition	Data type and values	Multiplicity and use
Accept Versions	Prioritized sequence of one or more standard versions accepted by client, with preferred versions listed first	Sequence of Character String type, each not empty Value is list of x.y.z ―version‖ values. SHALL contain "1.0.0"	Zero or one (optional) When omitted, return latest supported version

7.13. Keywords

NSG Requirement 15: **An NSG WMTS server shall provide a keyword list defined by Normative Reference 3, the NSG Metadata Foundation (NMF) Table 12 – Keywords (shown below in Table 15).**

Table 15: Keywords

Name / Role name	Definition	Obligation / Condition	Maximum occurrence	Data type / Domain	Business Rule
MD_Keywords	keywords, their type and reference source	Use obligation from referencing object	Use maximum occurrence from referencing object	Aggregated Class (MD_Identification)	
keyword	commonly used word(s) or formalized word(s) or phrase(s) used to describe the subject	M	N	CharacterString / Free text	
type	subject matter used to group similar keywords	O	1	Class / MD_KeywordTypeCode <<CodeList>> (KeywordTypeCode)	
thesaurusName	name of the formally registered thesaurus or a similar authoritative source of keywords	M	1	Class / CI_Citation <<DataType>>	Note: this element is optional in NMF

7.14. Cacheable Resources

NSG Requirement 16: An NSG WMTS server shall provide expiration date using HTTP control headers to support caching information.

Provision of expiration information is important because it improves the efficiency of client caching thereby reducing the WMTS server load and ultimately reducing the load time for users. To enable efficient web caching, an NSG WMTS server shall include expiration date in the server responses.

Caching works by marking certain data as being needed to be updated at different intervals. For example, information on a 1:2,000,000 scale Jet Navigation Chart (JNC) is relatively static, and is unlikely to change from one month to the next. By caching the tiles associated with the JNC, the client is able to only download this information as needed, e.g., once a year. By the server telling the client to store these files and not download them when revisiting the geographic area, display performance is improved and network bandwidth is conserved. Conversely, for information that may change rapidly, such as weather data, the update interval may be set to expire in a very short timeframe, possibly on the order of minutes.

Internally caching is achieved by means of the proper HTTP control headers:
HTTP 1.0, uses the "Expires" header. This header indicates an expiration date. If your data is guaranteed to be static, or you know when the data is going to be updated, you can use a convenient future date in the Expires header.

HTTP 1.1, uses the "Cache-control" header. This header indicates a period of time to cache the data before expiration. If your data is guaranteed to be static, or you know when the data is going to be updated, you can use a convenient period of time in the Cache-control header.

Client caching capabilities will vary greatly and should be designed around a number of considerations to include data latency, currency requirement, area of interest coverage requirements, storage capacity, and network connectivity.

7.15. Recommendations for Implementation

NSG Recommendation 1: If a service requires feature coordinate information, it is recommended that FeatureInfo documents be offered in the MIME type format "application/gml+xml; version=3.2".

NSG Recommendation 2: Any possible order of the variables and values in the URL template is valid. Nevertheless the following order is recommended:

1. style
2. firstDimension
3. lastDimension
4. TileMatrixSet
5. TileMatrix
6. TileRow
7. TileCol
8. J and I

NSG Recommendation 3: The Themes section of an NSG WMTS service metadata document should contain data about how layers are organized thematically.

In the Contents section of WMTS, layers are represented as a linear list without hierarchy, and a hierarchy of themes is specified separately in the Themes section, removing the need to specify complex inheritance rules for layer properties. This separates both concepts and makes it easy for a client to ignore the theme hierarchy or even to force another layer organization. Also it allows servers to offer more than one layer organization (in more than one themes section).
Each theme has a human-readable description (i.e., a title) and a list of layer references and child themes. It is possible for a layer to be a member of more than one theme, and for a layer to exist without being a member of any theme.

The normative reference for designation of theme names is the NAS (Normative Reference 4). Appropriate Theme names are to be determined from the NAS at the Principle View Group level, e.g., Transportation, or at the NAS View level, i.e., Inland Water Transportation and Associated Features, Railways and Associated Features, etc.

NSG Recommendation 4: NSG WMTS clients should be have the ability to support transparency.

Client support of transparency is important to be able to overlay tiles on top of other map data for the same geographical area.

8. NSG Queryable WMTS

8.1. Normative Requirements

The Normative requirements requested by this conformance class are summarized in Table 16.

Table 16: NSG Queryable WMTS Normative Server Requirements

No.	Category	Modifier	NSG Requirement	Compliance	Section
17	Service Type	N/A	A Queryable NSG server shall be compliant to the NSG Basic and Queryable WMTS conformance classes.	M	8.2
18	GetCapabilities Response (Conformance Class)	N/A	A Queryable NSG WMTS server shall include the following information in the abstract element of the service metadata: "This service implements the NSG WMTS Interoperability Standard version 1.0.0, Queryable conformance class."	M	8.3
19	GetFeatureInfo	KVP	A Queryable NSG WMTS server shall implement HTTP GET transfer of the GetFeatureInfo operation request using KVP encoding.	M	8.4
20		SOAP	A Queryable NSG WMTS server shall implement SOAP encoding using HTTP POST transfer of the GetFeatureInfo operation request, using SOAP version 1.2 encoding.	O	8.4
21		REST	A Queryable NSG WMTS server shall provide standard endpoints from which representation of the FeatureInfo resources can be obtained.	M	8.4
22	Output Format (GetFeatureInfo)	N/A	A Queryable NSG WMTS server shall provide the GetFeatureInfo output format in text/XML and text/HTML.	M	8.5

8.2. Service Type

According to this Interoperability Standard, a Queryable WMTS includes all Basic WMTS operations and an additional GetFeatureInfo operation.

NSG Requirement 17: An NSG Queryable WMTS server shall be compliant to the Basic and Queryable WMTS conformance classes.

8.3. GetCapabilities Operation Response

In response to a GetCapabilities request, an NSG WMTS server shall use the abstract element of the service metadata to identify the conformance class supported.

NSG Requirement 18: An NSG WMTS server shall include the following information in the abstract element of the service metadata: "This service implements the NSG WMTS Interoperability Standard version 1.0.0, Queryable conformance class."

8.4. GetFeatureInfo Requests

NSG Requirement 19: An NSG WMTS server shall implement HTTP GET transfer of the GetFeatureInfo operation request using KVP encoding.

KVP encoding of the GetFeatureInfo operation request shall follow the requirement for operation parameters specified in Table 17 and that follows the abstract description specified in Normative Reference 1, Table 25. NSG WMTS servers shall support KVP requests for representations of image tiles by declaring support for and correctly handling GetTile requests.

Table 17: GetFeatureInfo operation request URL parameters

Request Parameter	**OGC Mandatory/ Optional**	**NSG Mandatory/ Optional**	**Mandated Value**	**Definition and format**
Service	M	M	WMTS	Service type identifier.
Request	M	M	GetFeatureInfo	Operation name.
Version	M	M	1.0.0	Standard and schema version for this operation
Sample dimensions	O	O		Value allowed for this dimension.
Layer, Style, Format, *Sample dimensions,* TileMatrixSet, TileMatrix, TileRow, and TileCol	O Use of these parameters SHALL match those in the corresponding GetTile request described in Table 29, Normative Reference 1	O		The values of these parameters SHALL match those in the corresponding GetTile request described in Table 29, Normative Reference 1.

Request Parameter	OGC Mandatory/ Optional	NSG Mandatory/ Optional	Mandated Value	Definition and format
Layer, Style, Format, *Sample dimensions,* TileMatrixSet, TileMatrix, TileRow, and TileCol	O Use of these parameters SHALL match those in the corresponding GetTile request described in Table 29, Normative Reference 1.	O		The values of these parameters SHALL match those in the corresponding GetTile request described in Table 29, Normative Reference 1.
J	M	M		Row index of a pixel in the
I	M	M		Column index of a pixel in the tile.
InfoFormat	M	M		Output format of the retrieved information.

NSG Requirement 20: An NSG WMTS server shall implement SOAP encoding using HTTP POST transfer of the GetFeatureInfo operation request, using SOAP version 1.2 encoding.

The response of a successful SOAP-encoded GetFeatureInfo operation request shall be a document with the MIME Type specified by the InfoFormat parameter of the request wrapped in the SOAP version 1.2 envelope.

NSG Requirement 21: An NSG WMTS server shall provide standard endpoints from which representation of the FeatureInfo resources can be obtained.

The ServiceMetadata document contains a list of Layer Elements (Table 11) and each layer that is available to be retrieved and is queryable shall have one or more <ResourceURL> elements with the “resourceType” attribute set to “FeatureInfo” and a template attribute. The template attribute contains a URL template that can be converted to a URL by using a template processor. The FeatureInfo is obtained in the format specified by the attribute “format” by requesting the resource with a standard HTTP GET.

8.5. Output Format (GetFeatureInfo)

NSG Requirement 22: An NSG WMTS server shall provide the GetFeatureInfo output format in text/XML and text/HTML.

Text/HTML is required to support services employing mainstream, generic browsers and/or light clients.

9. Client Requirements and Recommendations

Normative WMTS client requirements are summarized in Table 18.

Table 18: Normative WMTS Client Requirements

No.	Category	Modifier	NSG Requirement	Compliance
1	Architectural Patterns/Encoding	N/A	An NSG WMTS client shall support REST and KVP.	M
2	GetCapabilities	KVP	An NSG WMTS client shall support a GetCapabilities operation by sending a GET or POST HTTP message with the "request" parameter set to "GetCapabilities.	M
3		SOAP	An NSG WMTS client shall be capable of issuing SOAP encoded GetCapabilities operation requests.	O
4		REST	An NSG WMTS client shall issue a GetResourceRepresentation request to access a Service Metadata document by requesting a file using the URL.	M
5	GetTile	KVP	An NSG WMTS client shall support KVP encoding of the GetTile operation request using parameters specified in Normative Reference 1, Table 29 (GetTile operation request URL parameters).	M
6		SOAP	An NSG WMTS client shall support SOAP encoding using HTTP POST transfer of the GetTile operation request, using SOAP 1.2.	O
7		REST	An NSG WMTS client shall request a tile representation using a tile URL.	M
8	GetFeatureInfo	KVP	An NSG WMTS client shall provide the capability to obtain more information for a pixel (I,J) on a particular tile by using data parameters as specified in Normative Reference 1, Figure 11 and Table 25. Only if implementing a Queryable WMTS.	C
9		SOAP	An NSG WMTS client shall provide the capability to obtain more information for a pixel (I,J) on a particular tile by SOAP encoding the GetFeatureInfo request. Only if implementing a Queryable WMTS.	C
10		REST	An NSG WMTS client shall have the capability to query those layers with one or more <ResourceURL> elements with the "resourceType" attribute set to "FeatureInfo" and a template attribute. Only if implementing a Queryable WMTS.	C
11	Coordinate Reference Systems	N/A	An NSG WMTS client shall support the World Geodetic System WGS 84 expressed as: • Geographic latitude, then longitude, recorded in decimal degrees - "EPSG::4326"	M

No.	Category	Modifier	NSG Requirement	Compliance
			(preferred) or • Geographic longitude, then latitude recorded in decimal degrees - "CRS:84"	
12	Projections	N/A	An NSG WMTS client shall support the following projections whose validity zones overlap data published by the service: • World Mercator Projection - "EPSG::3395" • UPS projection over WGS84 (north zone) - "EPSG::5041" • UPS projection over WGS84 (south zone) - "EPSG::5042"	M
13	Well-Known Scale Sets	N/A	An NSG WMTS client shall support the Well-Known Scale Sets identified in Annex C.	M
14	Tile Formats	N/A	An NSG WMTS client shall support image/png, image/jpeg, and image/gif file formats.	M
15	Negotiation of Standard Version	KVP	An NSG WMTS client shall support negotiation of the standard version used for client-server interactions.	M
16		SOAP	An NSG WMTS client shall support negotiation of the standard version used for client-server interactions.	O
17	Key Word List	N/A	An NSG WMTS client shall support a keyword list based upon the NSG Metadata Foundation (NMF).	M
18	Cacheable Resources	N/A	An NSG WMTS client shall support caching information (expiration date) for the data.	M
19	Output Format (GetFeatureInfo)	N/A	An NSG WMTS client shall support the GetFeatureInfo output format in text/XML and text/HTML. Only if implementing a Queryable WMTS.	C
20	Transparency	N/A	An NSG WMTS client should have the capability to support transparency. (Client support of transparency is important to be able to overlay tiles on top of other map data for the same geographical area).	O

Annex A – NSG OCG WMTS Abstract Test Suite (Normative)

A.1 Introduction

This abstract test suite specifies at a high level how server and client implementations of this standard SHALL be tested for conformance to this Interoperability Standard and the parent standard (Normative Reference 1). The framework for such abstract test suites is specified in ISO 19105: Geographic information – Conformance and testing, especially Clauses 7 and 9.

An abstract test suite contains multiple abstract tests, grouped into one or more test modules. This abstract test suite consists of two top-level test modules:

a) Server test module – Abstract tests for checking conformance of server implementations with the normative requirements of this Interoperability Standard. Server test modules are aligned according to the NSG Basic WMTS Conformance Class (Section A.2) and NSG Queryable WMTS Conformance Class (Section A.3).

b) Client test module – Abstract tests for checking conformance of client implementations with the normative requirements of this Interoperability Standard. Development of Client test modules are identified as a future work item in Section iv.

In the client and server test modules, all operations specified and implemented SHALL be tested, including KVP HTTP GET, and SOAP HTTP POST transfer and RESTFul HTTP GET transfer of each operation request. In the standard test module, all operations specified SHALL be checked, including KVP HTTP GET, SOAP HTTP POST and RESTFul HTTP GET transfer of operation requests. All operation request and response parameters specified or implemented SHALL be tested. Of course, some operations, transfer methods, and parameters are specified as optional implementation by servers. Any optional item not implemented by a server SHALL not be tested. Also, items not implemented by a client SHALL not be tested.

A.2 Server test module (NSG Basic WMTS)

NSG Requirement 1: An NSG server shall be compliant to the NSG Basic WMTS conformance class.

a) Test Purpose:

Verify that an NSG server satisfies all requirements for a Basic WMTS conformance class.

b) Test Method:

Submit a GetCapabilities and a GetTile request to the server and verify that it is providing proper responses.

c) References:

Requirements 2, 3, 5, 6, 8, 9, 10, 11, 12, 13, 15, 16

d) Test Type:

Capability

NSG Requirement 2: An NSG WMTS server shall support REST and KVP.

a) Test Purpose: Verify that a server is capable of handling procedure-oriented architectural style through KVP encoding as well as Resource Oriented Architectural style through REST encoding.

b) Test Method: This requirement is satisfied upon successful completion of testing for the following Requirements: 3, 5, 6, 8, and 13.

c) Reference: Section 7.

d) Test Type: Capability

NSG Requirement 3: An NSG WMTS server shall declare its support for GetCapabilities operations using KVP with HTTP GET.

a) Test Purpose: Verify that a server accepts HTTP GET transferred requests for GetCapabilities operation.

b) Test Method: Submit HTTP GET transferred requests for each operation. Verify that the server accepts and responds to these requests as specified and implemented. Check that the server accepts at least one HTTP GET transfer of requests for GetCapabilities operation.

c) Reference: Section 7.6.

d) Test Type: Capability

NSG Requirement 4: An NSG WMTS server shall generate a ServiceMetadata document in response to a SOAP encoded GetCapabilities request.

a) Test Purpose: Verify that a server generates a ServiceMetadata document in response to a SOAP encoded GetCapabilities request.

b) Test Method: Execute HTTP POST transfer of the GetCapabilities operation request using SOAP version 1.2. Verify that the server accepts and responds to this request as specified and implemented.

c) Reference: Section 7.6.

d) Test Type: Capability

NSG Requirement 5 Part 1: An NSG WMTS server shall generate a ServiceMetadata document in response to a GetResourceRepresentation request in REST architecture.

a) Test Purpose: Verify that an NSG WMTS server satisfies all Service Metadata requirements for a GetCapabilities operation request (procedure oriented architectural style) or a resource request (resource oriented architectural style) (Requirements 3 and 5).

b) Test Method: Submit a GetCapabilities or operation request and verify the following:
   1) The response has all required service metadata elements.
   2) The response uses <AccessContraints> to identify classification levels for the service.
   3) The response in the Abstract element contains the following information: "This service implements the NSG WMTS 1.0.0 InteroperabilityStandard Version 1.1, Basic conformance class." If Queryable conformance class is implemented, the response reads as follows: "This service implements the NSG WMTS Interoperability Standard version 1.0.0, Queryable conformance class."
   4) The response provides keywords based upon the NMF for the following: Layer data structure, Style data structure, Dimension data structure, TileMatrixSet data structure, TileMatrix data structure, and Themes data structure.
   5) The response provides information on the supported styles.
   6) The response provides a defined style for the default style.
   7) The response provides an associated legend in at least one of the following formats: PNG, GIF, JPEG.
   8) The provided LegendURL is accessible online.
   9) The response provides Dimension information, if applicable to a layer.
   10)The response provides a valid output format for Layer data structure infoFormat parameter thus enabling GetFeatureInfo.
   11)The response provides scale denominators for all layers.
   12)The provided <MinScaleDenominator> value is less than or equal to the <MaxScaleDenominator>.
   13)The list of features is resolvable through the provided FeatureListURLs.
   14)The data is resolvable through the provided URL for all provided DatURLs.

c) Reference: Section 7.6.

d) Test Type: Capability

NSG Requirement 5 Part 2: An NSG WMTS server shall generate a ServiceMetadata document in response to a GetResourceRepresentation request in REST architecture.

a) Test Purpose: Verify that a server handles a URL service metadata request.

b) Test Method: Request a Service metadata URL and other URL resources using correct and incorrect URLs. Verify that the server respond with the right resource to correct URLs, and a returns HTTP errors for invalid URLs.

c) Reference: Section 7.6.

d) Test Type: Capability

NSG Requirement 6: An NSG WMTS server shall respond to a GetTile operation request with a tile map that complies with the requested parameters.

a) Test Purpose: Verify that for a KVP encoded GetTile operation, for each GetTile format, when the Format parameter is set to that format, the MIME type of the response matches that format.

b) Test method: Pass if for each GetTile format, when the Format parameter is set to that format, the MIME type of the response matches that format.

c) Reference: Section 7.7.

d) Test Type: Capability

NSG Requirement 7: An NSG WMTS server shall respond to a SOAP encoded GetTile operation request with an image in the MIME type specified by the Format parameter of the request.

a) Test Purpose: Verify that for a SOAP encoded GetTile Operation, for each GetTile format, when the Format parameter is set to that format, the MIME type of the response matches that format.

b) Test method: Pass if for each GetTile format, when the Format parameter is set to that format, the MIME type of the response matches that format.

c) Reference: Section 7.7.

d) Test Type: Capability

NSG Requirement 8: An NSG WMTS server shall provide standard endpoints from which a representation of each Tile resource can be obtained.

a) Test Purpose: Verify that a server handles a URL service metadata request.

b) Test Method: Request a Service metadata URL and other URL resources using correct and incorrect URLs. Verify that the server respond with the right resource to correct URLs, and a returns HTTP errors for invalid URLs.

c) Reference: Section 7.7.

d) Test Type: Capability

NSG Requirement 9: An NSG WMTS server shall support the World Geodetic System WGS 84 expressed as:

- Geographic latitude, then longitude, recorded in decimal degrees (preferred) - "EPSG::4326" or
- Geographic longitude, then latitude recorded in decimal degrees - "CRS:84"

a) Test Purpose: Verify that an NSG WMTS server satisfies all the requirements for handling coordinate reference systems (NSG Requirement 9).

b) Test Method:
   1) Submit requests and verify that all supported coordinate reference systems are advertised for all available data in the XML response (Capabilities document) to a valid GetCapabilities request.
   2) Submit GetTile requests and verify that tiles are provided in appropriate projections for each validity zone.

c) Reference: Section 7.8.

d) Test Type: Capability

NSG Requirement 10: An NSG WMTS server shall support the following projections whose validity zones overlap data published by the service:

- World Mercator Projection - "EPSG::3395"
- UPS projection over WGS84 (north zone) - "EPSG::5041"
- UPS projection over WGS84 (south zone) - "EPSG::5042"

NSG WMTS Projections

a) Test Purpose: Verify that an NSG WMTS server satisfies all the requirements for handling projections (NSG Requirement 10).

b) Test Method:
   1) Submit requests and verify that all supported CRS are advertised for all available data in the XML response (Capabilities document) to a valid GetCapabilities request.
   2) Submit GetTile requests and verify that tiles are provided in appropriate projections for each validity zone.

c) c) Reference: Section 7.9.

d) d) Test Type: Capability

NSG Requirement 11: An NSG WMTS server shall employ the Well-Known Scale Sets identified in Annex C (based upon World Mercator projection EPSG::3395, WGS 84 EPSG::4326 and UPS EPSG::5041, EPSG::5042).

a) Test purpose: Verify that the WellKnownScaleSets identified in Annex C are employed.

b) Test method: Submit requests and verify that the WellKnownScaleSet that is advertised conforms to the tiling scheme identified in Annex C. That is the different ScaleDenominators with values starting from the largest scale denominator in the WellKnownScaleSet table and all intermediate scale denominators are represented.

c) Reference: Section 7.10.

d) Test type: Capability

NSG Requirement 12: An NSG WMTS server shall be capable of offering tiles in the image/png, image/jpeg, and image/gif file formats.

a) Test Purpose: Verify that an NSG WMTS server satisfies all the requirements for supported outputs.

b) Test Method: Submit requests and verify that the server implements support for:

    1) Submit a GetCapabilities request (FORMAT = text/html) and verify that the response is text/html
    2) Submit a GetFeatureInfo request (FORMAT = text/html) and verify that the response is text/html
    3) A response to a "GetCapabilities" request in English language
    4) Submit a GetTile request (FORMAT = image/png) and verify that the response is image/png
    5) Submit a GetTile request (FORMAT = image/gif) and verify that the response is image/gif
    6) Submit a GetTile request (FORMAT = image/jpeg) and verify that the response is image/jpeg.

c) Reference: Subclause 7.3.2 of OWS Common [OGC 06-121r3].

d) Test Type: Capability

NSG Requirement 13: An NSG WMTS server shall support negotiation of the standard version used for client-server interactions with KVP encoding.

a) Test Purpose: Verify that a server satisfies the requirements for version negotiation.

b) Test Method: Submit GetCapabilities operation requests containing version numbers lower than, higher than, and equal to the version supported by the server. Verify that the server responses are in accord with the specified rules for version negotiation.

c) Reference: Subclause 7.3.2 of OWS Common [OGC 06-121r3].

d) Test Type: Capability

NSG Requirement 14: An NSG WMTS server shall support negotiation of the standard version used for client-server interactions with SOAP encoding.

a) Test Purpose: Verify that a server satisfies the requirements for version negotiation.

b) Test Method: Submit GetCapabilities operation requests containing version numbers lower than, higher than, and equal to the version supported by the server. Verify that the server responses are in accord with the specified rules for version negotiation.

c) Reference: Subclause 7.3.2 of OWS Common [OGC 06-121r3].

d) Test Type: Capability

NSG Requirement 15: An NSG WMTS server shall provide a keyword list based upon the NSG Metadata Foundation (NMF).

a) Test Purpose: Verify that a server satisfies the requirements for use of the NSG Metadata Foundation.

b) Test Method: Submit a GetCapabilities request and verify that the response provides a keywordlist that is based on the NMF, Version 3.0.

c) Reference: Section 7.13.

d) Test Type: Capability

NSG Requirement 16: An NSG WMTS server shall provide caching information (expiration date) for the data.

a) Test Purpose: Verify that a server provides caching information for the data.

a) Test Method: Submit a GetTile request and verify that the response provides caching information for tile data returned by the server.

b) Reference: Section 7.14.

c) Test Type: Capability

A.3 Server test module (NSG Queryable WMTS)

NSG Requirement 17: An NSG Queryable WMTS server shall be compliant to the NSG WMTS Basic and Queryable WMTS conformance classes.

a) Test Purpose: Verify that an NSG Queryable WMTS server satisfies all requirements for an NSG WMTS Basic (Requirement 1-16) and an NSG Queryable WMTS (Requirement 17-22).

b) Test Method: See Annex A.2 and A.3.

c) Reference: N/A

d) Test Type: Capability

NSG Requirement 18: An NSG WMTS server shall include the following information in the abstract element of the service metadata: "This service implements the NSG WMTS Interoperability Standard version 1.0.0 , Queryable conformance class".

a) Test Purpose: Verify that an NSG Queryable WMTS interface satisfies requirement 18 by adding the correct text to the abstract in the GetCapabilitiesResponse.

b) Test Method: Submit a GetCapabilities request and verify that the response in the Abstract element contains the following information: "This service implements the NSG WMTS Interoperability Standard version 1.0.0, Queryable conformance class".

c) Reference: Section 8.3.

d) Test Type: Capability

NSG Requirement 19: An NSG WMTS server shall implement HTTP GET transfer of the GetFeatureInfo operation request using KVP encoding.

a) Test Purpose: Verify that an NSG WMTS interface satisfies all requirements for the operation GetFeatureInfo.

b) Test Method: Submit a GetFeatureInfo request from the client and verify that an appropriate response is returned according to the requested InfoFormat.

c) Reference: Section 8.4.

d) Test Type: Capability

NSG Requirement 20: An NSG WMTS server shall implement SOAP encoding using HTTP POST transfer of the GetFeatureInfo operation request, using SOAP version 1.2 encoding.

a) Test Purpose: Verify that an NSG WMTS interface satisfies all requirements for the operation GetFeatureInfo.

b) Test Method: Submit a GetFeatureInfo request from the client and verify that an appropriate response is returned according to the requested InfoFormat.

c) Reference: Section 8.4.

d) Test Type: Capability

NSG Requirement 21: An NSG WMTS server shall provide standard endpoints from which representation of the FeatureInfo resources can be obtained.

a) Test Purpose: Verify that a server satisfies RESTful requests.

b) Test Method: Verify that the server provides a Service Metadata document that includes complete ResourceURL information with resourceType=tile on Layer section if tiles of this layer are able for RESTful.

c) Reference: Section 8.4.

d) Test Type: Capability

NSG Requirement 22: An NSG WMTS server shall provide the GetFeatureInfo output format in text/XML and text/HTML.

a) Test Purpose: Verify that a server provides GetFeatureInfo in a text/XML or text/HTML format.

b) Test Method: Submit a GetFeatureInfo request and verify that the response is in text/XML or text/HTML format.

c) Reference: Section 8.5.

d) Test Type: Capability

Annex B – Implementation Conformance Statement

	NSG WMTS 1.0.0 Interoperability Standard Version 1.1 Server Implementation Conformance Statement (ICS) B=Basic WMTS Q=Queryable WMTS I=Implemented P/F=Pass/Fail M=Mandatory O=Optional C=Conditional Implementation Under Test: Conformance Class: Basic__ Queryable__ Test Point: Test Sponsor: Date of Test Completion: Test Organization:					
Conformance Class	**Category**	**Parameter**	**Obligation**			
			B	**Q**	**I**	**P/F**
NSG Basic WMTS Those aspects of Web Map Tile Service server behavior that are required by the Interoperability Standard	Service Type	Complies with the NSG Basic WMTS conformance class	M	M		
	Architectural Patterns/Encoding	Supports REST and KVP	M	M		
		Supports SOAP	O	O		
	GetCapabilities	Supports GetCapabilities operations using KVP with HTTP GET by providing an OperationsMetadata section in the ServiceMetadata document with an Operation section for each supported HTTP request type.	M	M		
		Generates a ServiceMetadata document, in response to a SOAP encoded GetCapabilities request, that looks like the one in Normative Reference 1, section 7.1.1.3, wrapped in a SOAP 1.2 envelope.	O	O		
		Generates a ServiceMetadata document in response to a GetResource Representation request in REST architecture that looks like the one described in section 7.1.1.3 of Normative Reference 1.	M	M		
	GetTile	Responds to a GetTile operation request encoded in KVP with a tile map that complies with the requested parameters.	M	M		

	NSG WMTS 1.0.0 Interoperability Standard Version 1.1 Server Implementation Conformance Statement (ICS) **B=Basic WMTS Q=Queryable WMTS I=Implemented P/F=Pass/Fail** **M=Mandatory O=Optional C=Conditional** **Implementation Under Test:** **Conformance Class: Basic__ Queryable__** **Test Point:** **Test Sponsor:** **Date of Test Completion:** **Test Organization:**					
		Responds to a SOAP-encoded GetTile operation request with an image in the MIME type specified by the Format parameter of the request, wrapped in the SOAP version 1.2.	O	O		
		Provides standard endpoints in REST architecture from which a representation of each Tile resource can be obtained.	M	M		
	Coordinate Reference Systems	Supports the World Geodetic System WGS 84 expressed as: · Geographic latitude, then longitude, recorded in decimal degrees - “EPSG::4326” (preferred) or · Geographic longitude, then latitude recorded in decimal degrees - “CRS:84”	M	M		
	Projections	An NSG WMTS server shall support the following projections whose validity zones overlap data published by the service: · World Mercator Projection - “EPSG::3395” · UPS projection over WGS84 (north zone) - “EPSG::5041” · UPS projection over WGS84 (south zone) - “EPSG::5042”	M	M		
	Well-Known Scale Sets	Supports Well-Known Scale Sets identified in Annex C.	M	M		

NSG WMTS 1.0.0 Interoperability Standard Version 1.1 Server Implementation Conformance Statement (ICS)

B=Basic WMTS Q=Queryable WMTS I=Implemented P/F=Pass/Fail
M=Mandatory O=Optional C=Conditional

Implementation Under Test:

Test Point:

Date of Test Completion:

Conformance Class:
Basic__ Queryable__

Test Sponsor:

Test Organization:

Conformance Class	Category	Parameter	Obligation			
			B	Q	I	P/F
NSG Basic WMTS Those aspects of Web Map Tile Service server behavior that are required by the Interoperability Standard	Tile Formats	Offers tiles in the following formats: - image/png - image/jpeg - image/gif	M	M		
	Negotiation of Standard Version	Supports negotiation of the standard version used for client-server interactions using KVP.	M	M		
		Supports negotiation of the standard version used for client server interactions using SOAP.	O	O		
	Key Word List	Provides a keyword list based upon the NSG Metadata Foundation (NMF).	M	M		
	Cacheable Resources	Provides caching information (expiration date) for the data.	M	M		
	MIME Type	If a service requires feature coordinate information, it is recommended that FeatureInfo documents be offered in the MIME type format “application/gml+xml; version=3.2”.	O	O		
	Order of Variables in URL Template	Any possible order of the variables and values in the URL template is valid. Nevertheless, recommend the following order: style, firstDimension, ..., lastDimension, TileMatrixSet, TileMatrix, TileRow, TileCol, J and I.	O	O		
	Themes	Recommend that the Themes section of a WMTS service metadata document contain data about how layers are organized thematically.	O	O		
	Transparency	Clients should have the ability to support transparency.	O	O		
NSG Queryable WMTS Those aspects of Web Map Tile Service server behavior that are required by the Interoperability Standard	Service Type	Complies with the NSG Basic and Queryable WMTS conformance classes.		M		
	GetCapabilities Response (Conformance Class)	Provides the following information in the abstract element of the service metadata: “This service implements the NSG WMTS Interoperability Standard version 1.0.0, Queryable conformance class.”		M		
	GetFeatureInfo	Implements HTTP GET transfer of the GetFeatureInfo operation request using KVP encoding.		M		
		Implements SOAP encoding using HTTP POST transfer of the GetFeatureInfo operation request, using SOAP version 1.2 encoding.		O		
		Provides standard endpoints in REST architecture from which representation of the FeatureInfo resources can be obtained.		M		
	Output Format (GetFeatureInfo)	Provides the GetFeatureInfo output format in text/XML and text/HTML.		M		

NSG WMTS 1.0.0 Interoperability Standard Version 1.1 Client Implementation Conformance Statement (ICS)

B=Basic WMTS Q=Queryable WMTS I=Implemented P/F=Pass/Fail
M=Mandatory O=Optional C=Conditional

Implementation Under Test: | Conformance Class: Basic__ Queryable__
Test Point: | Test Sponsor:
Date of Test Completion: | Test Organization:

Conformance Class	Category	Parameter	Obligation			
			B	Q	I	P/F
NSG Basic and Queryable WMTS Those aspects of Web Map Tile Service client behavior that are required by the Interoperability Standard	Architectural Patterns/Encoding	Supports REST and KVP.	M	M		
		Supports SOAP.	O	O		
	GetCapabilities	An NSG WMTS client shall support a GetCapabilities operation using KVP by sending a GET HTTP message with the “request” parameter set to “GetCapabilities.	M	M		
		An NSG WMTS client shall be capable of issuing SOAP encoded GetCapabilities operation requests.	M	M		
		An NSG WMTS client shall issue a GetResourceRepresentation request in REST architecture to access a Service Metadata document by requesting a file using the URL.	M	M		
	GetTile	An NSG WMTS client shall support KVP encoding of the GetTile operation request using parameters specified in Normative Reference 1, Table 29 (GetTile operation request URL parameters).	M	M		
		An NSG WMTS client shall support SOAP encoding using HTTP POST transfer of the GetTile operation request, using SOAP 1.2.	M	M		
		An NSG WMTS client shall request a tile representation in REST architecture using a tile URL.	M	M		
	GetFeatureInfo	An NSG WMTS client shall provide the capability using KVP to obtain more information for a pixel (I,J) on a particular tile by using data parameters as specified in Normative Reference 1, Figure 11 and Table 25. Only if implementing a Queryable WMTS.		C		
		An NSG WMTS client shall provide the capability to obtain more information for a pixel (I,J) on a particular tile by SOAP encoding the GetFeatureInfo request. Only if implementing a Queryable WMTS.		C		
		An NSG WMTS client shall have the capability to query those layers in REST architecture with one or more <ResourceURL> elements with the “resourceType” attribute set to “FeatureInfo” and a template attribute. Only if implementing a Queryable WMTS.		C		
	Coordinate Reference Systems	An NSG WMTS client shall support the World Geodetic System WGS 84 expressed as: · Geographic latitude, then longitude, recorded in decimal degrees - “EPSG::4326” (preferred) or · Geographic longitude, then latitude recorded in decimal degrees – “CRS:84”	M	M		

	NSG WMTS 1.0.0 Interoperability Standard Version 1.1 Client Implementation Conformance Statement (ICS) B=Basic WMTS Q=Queryable WMTS I=Implemented P/F=Pass/Fail M=Mandatory O=Optional C=Conditional Implementation Under Test: Conformance Class: Basic__ Queryable__ Test Point: Test Sponsor: Date of Test Completion: Test Organization:					
Conformance Class	**Category**	**Parameter**	**Obligation**			
			B	**Q**	**I**	**P/F**
<u>NSG Basic and Queryable WMTS</u> Those aspects of Web Map Tile Service <u>client</u> behavior that are required by the Interoperability Standard	Projections	An NSG WMTS client shall support the following projections whose validity zones overlap data published by the service: · World Mercator Projection - "EPSG::3395" · UPS projection over WGS84 (north zone) - "EPSG::5041" · UPS projection over WGS84 (south zone) - "EPSG::5042"	M	M		
	Well-Known Scale Sets	An NSG WMTS client shall support the Well-Known Scale Sets identified in Annex C	M	M		
	Negotiation of Standard Version	An NSG WMTS client shall use KVP to support negotiation of the standard version used for client-server interactions.	M	M		
		An NSG WMTS client shall use SOAP to support negotiation of the standard version used for client-server interactions.	O	O		
	Key Word List	An NSG WMTS client shall support a keyword list based upon the NSG Metadata Foundation (NMF).	M	M		
	Cacheable Resources	An NSG WMTS client shall support caching information (expiration date) for the data.	M	M		
	Output Format (GetFeatureInfo)	An NSG WMTS client shall support the GetFeatureInfo output format in text/XML and text/HTML. Only if implementing a Queryable WMTS.		C		
	Transparency	An NSG WMTS client should have the capability to support transparency. (Client support of transparency is important to be able to overlay tiles on top of other map data for the same geographical area).	O	O		

Annex C – Well-known Scale Sets (Normative)

C.1. World Mercator - “EPSG::3395”

The World Mercator “EPSG::3395” well-known scale set is defined in this NSG WMTS Implementation Interoperability Profile. To be conformant to this well-known scale set, a WMTS server SHALL allow responses from the largest scale denominator in Table 19 and all intermediate scale denominators down to the most detailed scale resolution of that data. It is therefore not required to support the smallest scale denominator in order to be conformant to a well-known scale set.

The World Mercator “EPSG::3395” WMTS tile scheme has a square layout.

The projection bounds for the World Mercator raster tile pyramid are (-20037508.342789244, -20037508.342789244) to (20037508.342789244, 20037508.342789244) meters which yields the corresponding longitude, latitude limits of (-180, -85.084059) to (180, 85.084059)

The zoom level 0 tile covers the whole bounds of the projection which yields an area of 40,075,016.68557849 meters by 40,075,016.68557849 meters.

As referenced in Normative Reference 1, section 6.1. a) "The scale denominator is defined with respect to a 'standardized rendering pixel size' of 0.28 mm × 0.28 mm (millimeters)”[5] and the WGS84 equatorial earth diameter.

Two hundred fifty six (256) pixels in the horizontal dimension are defined to be 71.68 mm or 0.07168 meters. It represents 40,075,016.68557849 meters of Earth distance. Thus, the scale is 40,075,016.68557849 / 0.07168 or 559082264.0287178.

At zoom level 0, the pixel size in meters is 40,075,016.68557849 / 256 = 156543.033928041.

Scales and pixel sizes for the 3395 projection match the GoogleMapsCompatible scale set, even though the projections are different. The actual tile contents will also differ slightly.

The zoom level scale set and matrix width and height for the World Mercator raster tile pyramid are defined in Table 9.

[5] Given that, from the server side, the size of the pixels of client display devices cannot be predetermined, WMTS uses a “standard” pixel size of 0.28mm that provides a “common” relationship between pixel size and scale. This can create situations where a server advertises tiles for a given scale that results in visualization at a different scale on a client’s display. For example, a server advertised scale of 1:10000 may result in ~1:35000 visualizations on an Apple retina display with a 0.078mm pixel size (0.28 is only a convenient "constant"). So, to show the “real” scale displayed in a retina device, the scale advertised by the server must be multiplied by ~3.6 (0.28/0.078).

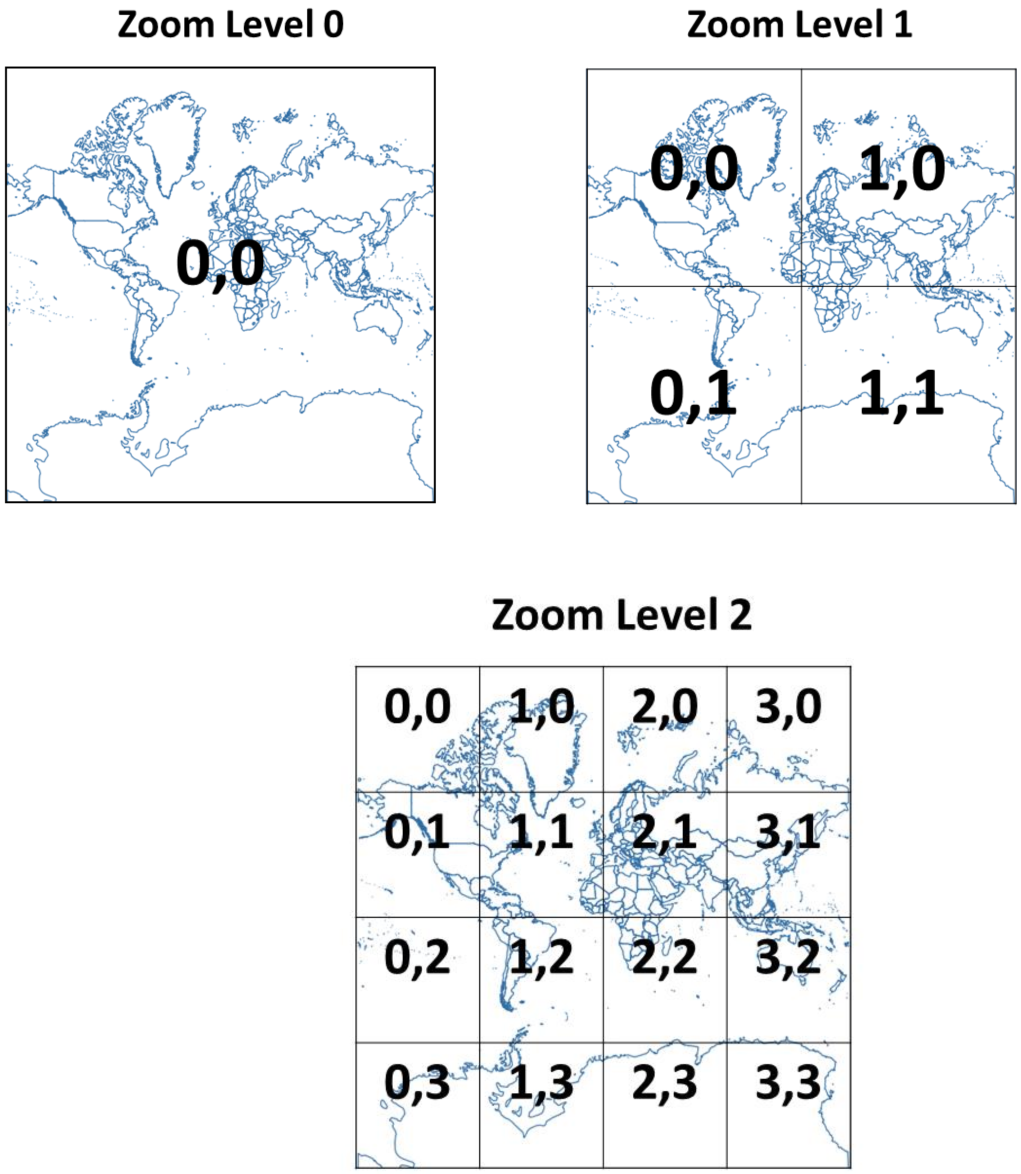
Table 19: Zoom Level Scale Set and Matrix Dimensions-EPSG::3395

Zoom Level	Scale	Pixel Size (m)	*matrix_width*	*matrix_height*
0	559082264.02871774553571428572	156543.03392804096875	1	1
1	279541132.01435887276785714286	78271.516964020484375	2	2
2	139770566.00717943638392857143	39135.7584820102421875	4	4
3	69885283.003589718191964285715	19567.87924100512109375	8	8
4	34942641.501794859095982142857	9783.939620502560546875	16	16
5	17471320.750897429547991071429	4891.9698102512802734375	32	32
6	8735660.375448714773995535714	2445.98490512564013671875	64	64
7	4367830.1877243573869977678571	1222.992452562820068359375	128	128
8	2183915.0938621786934988839286	611.4962262814100341796875	256	256
9	1091957.5469310893467494419643	305.74811314070501708984375	512	512
10	545978.77346554467337472098214	152.874056570352508544921875	1024	1024
11	272989.38673277233668736049107	76.4370282851762542724609375	2048	2048
12	136494.69336638616834368024553	38.21851414258812713623046875	4096	4096
13	68247.346683193084171840122768	19.109257071294063568115234375	8192	8192
14	34123.673341596542085920061386	9.554628535647031784057617188	16384	16384
15	17061.836670798271042960030692	4.7773142678235158920288085938	32768	32768
16	8530.918335399135521480015346	2.3886571339117579460144042969	65536	65536
17	4265.4591676995677607400076729	1.1943285669558789730072021484	131072	131072
18	2132.7295838497838803700038364	0.5971642834779394865036010742	262144	262144
19	1066.3647919248919401850019182	0.2985821417389697432518005371	524288	524288
20	533.18239596244597009250095928	0.1492910708694848716259002686	1048576	1048576
21	266.59119798122298504625047964	0.0746455354347424358129501343	2097152	2097152

22	133.29559899061149252312523964	0.0373227677173712179064750671	4194304	4194304
23	66.647799495305746261562620003	0.0186613838586856089532375336	8388608	8388608
24	33.323899747652873130781310002	0.0093306919293428044766187668	16777216	16777216

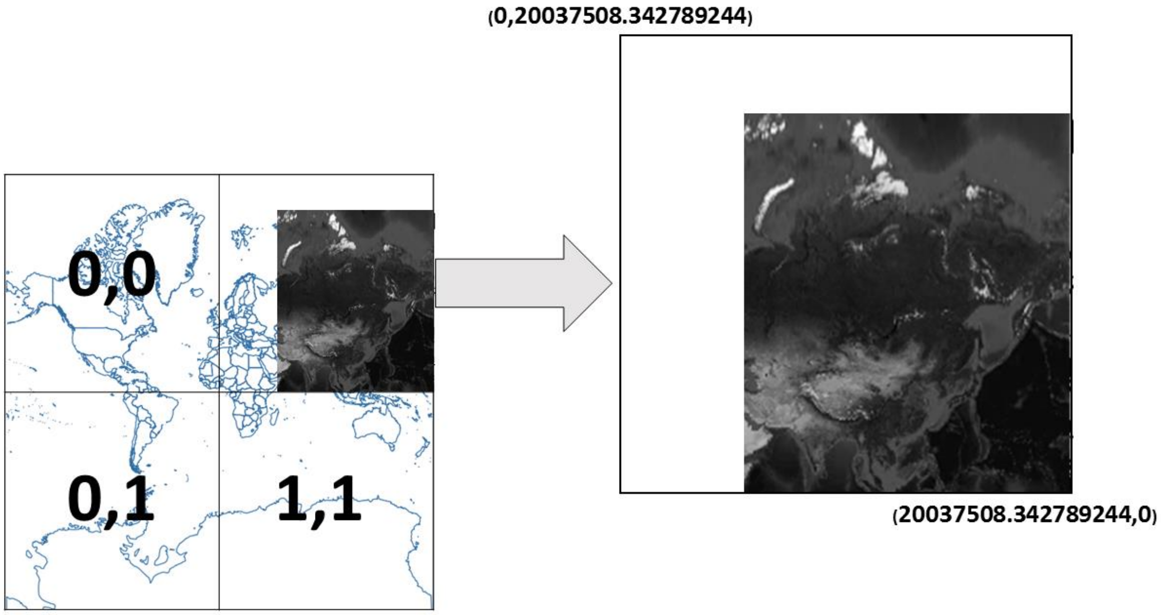
All tiles are **globally referenced** based on the zoom level and (column, row) index values on the global World Mercator grid. Figure 2 below shows the (column, row) index values for zoom levels 0 through 3.

Figure 2: World Mercator tile indexing



The bounding box should describe the maximum extent of the tiles. If source data does not completely fill a tile, pixels without data within the tile must be completely filled with a default value (e.g. transparency). The bounding box must be aligned with tile boundaries, not to data boundaries (Figure 3).

Figure 3: Bounding box and partial tile example



C.2. XML Description

```
<TileMatrixSet>
        <Identifier>EPSG3395TiledMercator</Identifier>
        <SupportedCRS>
                EPSG:3395
        </SupportedCRS>
        <BoundingBox crs="EPSG:3395">
                <LowerCorner>-2.0037508342789244E7 -2.0037508342789244E7
                </LowerCorner>
                <UpperCorner>2.0037508342789244E7 2.0037508342789244E7</UpperCorner>
        </BoundingBox>
        <TileMatrix>
                <Identifier>0</Identifier>
                <ScaleDenominator>5.59082264028718E8</ScaleDenominator>
                <TopLeftCorner>
                        -2.0037508342789244E7 2.0037508342789244E7
                </TopLeftCorner>
                <TileWidth>256</TileWidth>
                <TileHeight>256</TileHeight>
                <MatrixWidth>1</MatrixWidth>
                <MatrixHeight>1</MatrixHeight>
        </TileMatrix>
        <TileMatrix>
                <Identifier>1</Identifier>
                <ScaleDenominator>2.79541132014359E8</ScaleDenominator>
                <TopLeftCorner>
                        -2.0037508342789244E7 2.0037508342789244E7
                </TopLeftCorner>
                <TileWidth>256</TileWidth>
                <TileHeight>256</TileHeight>
                <MatrixWidth>2</MatrixWidth>
                <MatrixHeight>2</MatrixHeight>
        </TileMatrix>
        <TileMatrix>
                <Identifier>2</Identifier>
                <ScaleDenominator>1.397705660071795E8</ScaleDenominator>
                <TopLeftCorner>
                        -2.0037508342789244E7 2.0037508342789244E7
                </TopLeftCorner>
                <TileWidth>256</TileWidth>
                <TileHeight>256</TileHeight>
                <MatrixWidth>4</MatrixWidth>
                <MatrixHeight>4</MatrixHeight>
        </TileMatrix>
        <TileMatrix>
                <Identifier>3</Identifier>
                <ScaleDenominator>6.988528300358975E7</ScaleDenominator>
                <TopLeftCorner>
                        -2.0037508342789244E7 2.0037508342789244E7
                </TopLeftCorner>
        <TileWidth>256</TileWidth>
        <TileHeight>256</TileHeight>
        <MatrixWidth>8</MatrixWidth>
        <MatrixHeight>8</MatrixHeight>
</TileMatrix>
<TileMatrix>
        <Identifier>4</Identifier>
        <ScaleDenominator>3.4942641501794875E7</ScaleDenominator>
        <TopLeftCorner>
                -2.0037508342789244E7 2.0037508342789244E7
        </TopLeftCorner>
        <TileWidth>256</TileWidth>
        <TileHeight>256</TileHeight>
        <MatrixWidth>16</MatrixWidth>
        <MatrixHeight>16</MatrixHeight>
</TileMatrix>
<TileMatrix>
        <Identifier>5</Identifier>
        <ScaleDenominator>1.7471320750897437E7</ScaleDenominator>
        <TopLeftCorner>
                -2.0037508342789244E7 2.0037508342789244E7
        </TopLeftCorner>
        <TileWidth>256</TileWidth>
        <TileHeight>256</TileHeight>
        <MatrixWidth>32</MatrixWidth>
        <MatrixHeight>32</MatrixHeight>
</TileMatrix>
<TileMatrix>
        <Identifier>6</Identifier>
        <ScaleDenominator>8735660.375448719</ScaleDenominator>
        <TopLeftCorner>
                -2.0037508342789244E7 2.0037508342789244E7
        </TopLeftCorner>
        <TileWidth>256</TileWidth>
        <TileHeight>256</TileHeight>
        <MatrixWidth>64</MatrixWidth>
        <MatrixHeight>64</MatrixHeight>
</TileMatrix>
<TileMatrix>
        <Identifier>7</Identifier>
        <ScaleDenominator>4367830.187724359</ScaleDenominator>
        <TopLeftCorner>
                -2.0037508342789244E7 2.0037508342789244E7
        </TopLeftCorner>
        <TileWidth>256</TileWidth>
        <TileHeight>256</TileHeight>
        <MatrixWidth>128</MatrixWidth>
        <MatrixHeight>128</MatrixHeight>
</TileMatrix>
<TileMatrix>
        <Identifier>8</Identifier>
        <ScaleDenominator>2183915.0938621797</ScaleDenominator>
        <TopLeftCorner>
```

```
                -2.0037508342789244E7 2.0037508342789244E7
        </TopLeftCorner>
        <TileWidth>256</TileWidth>
        <TileHeight>256</TileHeight>
        <MatrixWidth>256</MatrixWidth>
        <MatrixHeight>256</MatrixHeight>
</TileMatrix>
<TileMatrix>
        <Identifier>9</Identifier>
        <ScaleDenominator>1091957.5469310898</ScaleDenominator>
        <TopLeftCorner>
                -2.0037508342789244E7 2.0037508342789244E7
        </TopLeftCorner>
        <TileWidth>256</TileWidth>
        <TileHeight>256</TileHeight>
        <MatrixWidth>512</MatrixWidth>
        <MatrixHeight>512</MatrixHeight>
</TileMatrix>
<TileMatrix>
        <Identifier>10</Identifier>
        <ScaleDenominator>545978.7734655449</ScaleDenominator>
        <TopLeftCorner>
                -2.0037508342789244E7 2.0037508342789244E7
        </TopLeftCorner>
        <TileWidth>256</TileWidth>
        <TileHeight>256</TileHeight>
        <MatrixWidth>1024</MatrixWidth>
        <MatrixHeight>1024</MatrixHeight>
</TileMatrix>
<TileMatrix>
        <Identifier>11</Identifier>
        <ScaleDenominator>272989.38673277246</ScaleDenominator>
        <TopLeftCorner>
                -2.0037508342789244E7 2.0037508342789244E7
        </TopLeftCorner>
        <TileWidth>256</TileWidth>
        <TileHeight>256</TileHeight>
        <MatrixWidth>2048</MatrixWidth>
        <MatrixHeight>2048</MatrixHeight>
</TileMatrix>
<TileMatrix>
        <Identifier>12</Identifier>
        <ScaleDenominator>136494.69336638623</ScaleDenominator>
        <TopLeftCorner>
                -2.0037508342789244E7 2.0037508342789244E7
        </TopLeftCorner>
        <TileWidth>256</TileWidth>
        <TileHeight>256</TileHeight>
        <MatrixWidth>4096</MatrixWidth>
        <MatrixHeight>4096</MatrixHeight>
</TileMatrix>
<TileMatrix>
        <Identifier>13</Identifier>
        <ScaleDenominator>68247.34668319311</ScaleDenominator>
        <TopLeftCorner>
                -2.0037508342789244E7 2.0037508342789244E7
        </TopLeftCorner>
        <TileWidth>256</TileWidth>
        <TileHeight>256</TileHeight>
        <MatrixWidth>8192</MatrixWidth>
        <MatrixHeight>8192</MatrixHeight>
</TileMatrix>
<TileMatrix>
        <Identifier>14</Identifier>
        <ScaleDenominator>34123.67334159656</ScaleDenominator>
        <TopLeftCorner>
                -2.0037508342789244E7 2.0037508342789244E7
        </TopLeftCorner>
        <TileWidth>256</TileWidth>
        <TileHeight>256</TileHeight>
        <MatrixWidth>16384</MatrixWidth>
        <MatrixHeight>16384</MatrixHeight>
</TileMatrix>
<TileMatrix>
        <Identifier>15</Identifier>
        <ScaleDenominator>17061.83667079828</ScaleDenominator>
        <TopLeftCorner>
                -2.0037508342789244E7 2.0037508342789244E7
        </TopLeftCorner>
        <TileWidth>256</TileWidth>
        <TileHeight>256</TileHeight>
        <MatrixWidth>32768</MatrixWidth>
        <MatrixHeight>32768</MatrixHeight>
</TileMatrix>
<TileMatrix>
        <Identifier>16</Identifier>
        <ScaleDenominator>8530.91833539914</ScaleDenominator>
        <TopLeftCorner>
                -2.0037508342789244E7 2.0037508342789244E7
        </TopLeftCorner>
        <TileWidth>256</TileWidth>
        <TileHeight>256</TileHeight>
        <MatrixWidth>65536</MatrixWidth>
        <MatrixHeight>65536</MatrixHeight>
</TileMatrix>
<TileMatrix>
        <Identifier>17</Identifier>
        <ScaleDenominator>4265.45916769957</ScaleDenominator>
        <TopLeftCorner>
                -2.0037508342789244E7 2.0037508342789244E7
        </TopLeftCorner>
        <TileWidth>256</TileWidth>
        <TileHeight>256</TileHeight>
        <MatrixWidth>131072</MatrixWidth>
        <MatrixHeight>131072</MatrixHeight>
</TileMatrix>
```

```
<TileMatrix>
        <Identifier>18</Identifier>
        <ScaleDenominator>2132.729583849785</ScaleDenominator>
        <TopLeftCorner>
                -2.0037508342789244E7 2.0037508342789244E7
        </TopLeftCorner>
        <TileWidth>256</TileWidth>
        <TileHeight>256</TileHeight>
        <MatrixWidth>262144</MatrixWidth>
        <MatrixHeight>262144</MatrixHeight>
</TileMatrix>
<TileMatrix>
        <Identifier>19</Identifier>
        <ScaleDenominator>1066.3647919248924</ScaleDenominator>
        <TopLeftCorner>
                -2.0037508342789244E7 2.0037508342789244E7
        </TopLeftCorner>
        <TileWidth>256</TileWidth>
        <TileHeight>256</TileHeight>
        <MatrixWidth>524288</MatrixWidth>
        <MatrixHeight>524288</MatrixHeight>
</TileMatrix>
<TileMatrix>
        <Identifier>20</Identifier>
        <ScaleDenominator>533.1823959624462</ScaleDenominator>
        <TopLeftCorner>
                -2.0037508342789244E7 2.0037508342789244E7
        </TopLeftCorner>
        <TileWidth>256</TileWidth>
        <TileHeight>256</TileHeight>
        <MatrixWidth>1048576</MatrixWidth>
        <MatrixHeight>1048576</MatrixHeight>
</TileMatrix>
<TileMatrix>
        <Identifier>21</Identifier>
        <ScaleDenominator>266.5911979812231</ScaleDenominator>
        <TopLeftCorner>
                -2.0037508342789244E7 2.0037508342789244E7
        </TopLeftCorner>
        <TileWidth>256</TileWidth>
        <TileHeight>256</TileHeight>
        <MatrixWidth>2097152</MatrixWidth>
        <MatrixHeight>2097152</MatrixHeight>
</TileMatrix>
<TileMatrix>
        <Identifier>22</Identifier>
        <ScaleDenominator>133.29559899061155</ScaleDenominator>
        <TopLeftCorner>
                -2.0037508342789244E7 2.0037508342789244E7
        </TopLeftCorner>
        <TileWidth>256</TileWidth>
        <TileHeight>256</TileHeight>
        <MatrixWidth>4194304</MatrixWidth>
```

```xml
        <MatrixHeight>4194304</MatrixHeight>
    </TileMatrix>
    <TileMatrix>
        <Identifier>23</Identifier>
        <ScaleDenominator>66.64779949530578</ScaleDenominator>
        <TopLeftCorner>
            -2.0037508342789244E7 2.0037508342789244E7
        </TopLeftCorner>
        <TileWidth>256</TileWidth>
        <TileHeight>256</TileHeight>
        <MatrixWidth>8388608</MatrixWidth>
        <MatrixHeight>8388608</MatrixHeight>
    </TileMatrix>
    <TileMatrix>
        <Identifier>24</Identifier>
        <ScaleDenominator>33.32389974765289</ScaleDenominator>
        <TopLeftCorner>
            -2.0037508342789244E7 2.0037508342789244E7
        </TopLeftCorner>
        <TileWidth>256</TileWidth>
        <TileHeight>256</TileHeight>
        <MatrixWidth>16777216</MatrixWidth>
        <MatrixHeight>16777216</MatrixHeight>
    </TileMatrix>
</TileMatrixSet>
```

C.3. WGS 84 lat/long - “EPSG::4326”

The WGS 84 lat/long “EPSG::4326” tile scheme has an equirectangular layout.

The projection bounds for the WGS 84 Geodetic raster tile pyramid are (-90, -180) to (90, 180)

Zoom level 0 will consist of two square tiles that cover the whole bounds of the projection.
As referenced in the OpenGIS® Web Map Tile Service Implementation Standard, OGC 07-057r7, Section 6.1. a) "The scale denominator is defined with respect to a 'standardized rendering pixel size' of 0.28 mm × 0.28 mm (millimeters)". This pixel size is nominal and is used to generate the meters per pixel values shown in Table 20 below.[6] At zoom level 0, the pixel size in degrees is 180/ 256 = 0.7031250.

The zoom level scale set and matrix width and height for the WGS 84 raster tile pyramid are defined below in Table 20. These tiles are fully compatible with the World CRS84 TileMatrixSet definition as specified in OGC 13-082r2 “WMTS Simple Profile”. Specifically, zoom levels 0-17 match the “WMTS Simple Profile” while zoom levels 18-23 provide additional flexibility for NSG users. Any discrepancies in the scale denominator and pixels size values shown below are simply an artifact of rounding, and do not reflect any difference in how the tiles are created or rendered.

Table 20: Zoom Level Scale Set and Matrix dimensions- WGS 84 lat/long

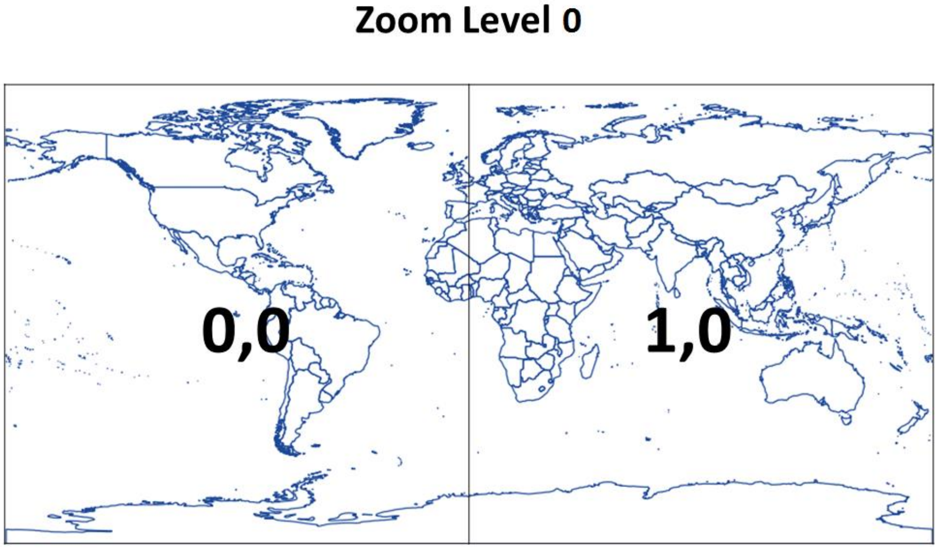
Zoom Level	Scale Denominator	Pixel Size	*matrix_width*	*matrix_height*
0	279541132.01435900	0.7031250	2	1
1	139770566.00717900	0.3515625	4	2
2	69885283.00358960	0.17578125	8	4
3	34942641.50179480	0.0878906250	16	8
4	17471320.75089740	0.0439453125	32	16
5	8735660.37544870	0.0219726563	64	32
6	4367830.18772435	0.0109863281	128	64
7	2183915.09386218	0.0054931641	256	128
8	1091957.54693109	0.0027465820	512	256

[6] Given that, from the server side, the size of the pixels of client display devices cannot be predetermined, WMTS uses a "standard" pixel size of 0.28mm that provides a "common" relationship between pixel size and scale. This can create situations where a server advertises tiles for a given scale that results in visualization at a different scale on a client's display. For example, a server advertised scale of 1:10000 may result in ~1:35000 visualizations on a display with a 0.078mm pixel size (0.28 is only a convenient "constant"). So, to show the "real" scale displayed on this display, the scale advertised by the server must be multiplied by ~3.6 (0.28/0.078).

Zoom Level	Scale Denominator	Pixel Size	*matrix_width*	*matrix_height*
9	545978.77346554	0.0013732910	1024	512
10	272989.38673277	0.0006866455	2048	1024
11	136494.69336639	0.0003433228	4096	2048
12	68247.34668319	0.0001716614	8192	4096
13	34123.67334160	0.0000858307	16384	8192
14	17061.83667080	0.0000429153	32768	16384
15	8530.91833540	0.0000214577	65536	32768
16	4265.45916770	0.0000107288	131072	65536
17	2132.72958385	0.0000053644	262144	131072
18	1066.36479192	0.0000026822	524288	262144
19	533.18239596	0.0000013411	1048576	524288
20	266.59119798	0.0000006706	2097152	1048576
21	133.29559899	0.0000003353	4194304	2097152
22	66.64779950	0.0000001676	8388608	4194304
23	33.32389975	0.0000000838	16777216	8388608

All tiles are **globally referenced** based on the zoom level and (column, row) index values on the global WGS 84 grid. Figure 4 below shows the (column, row) index values for zoom levels 1 through 3.

Figure 4: WGS 84 Tile Indexing



C.4. XML Description

```
<TileMatrixSet>
        <ows:Identifier>WGS1984Quad</ows:Identifier>
        <ows:SupportedCRS>urn:ogc:def:crs:EPSG::4326</ows:SupportedCRS>
        <BoundingBox crs="EPSG:4326">
                <LowerCorner>-90 -180</LowerCorner>
                <UpperCorner>90 180</UpperCorner>
        </BoundingBox>
        <TileMatrix>
                <ows:Identifier>0</Identifier>
                <ScaleDenominator>279541132.014359</ScaleDenominator>
                <TopLeftCorner>90 -180</TopLeftCorner>
                <matrix_width>2</matrix_width>
                <matrix_height>1</matrix_height>
        </TileMatrix>
        <TileMatrix>
                <ows:Identifier>1</Identifier>
                <ScaleDenominator>139770566.007179</ScaleDenominator>
                <TopLeftCorner>90 -180</TopLeftCorner>
                <matrix_width>4</matrix_width>
                <matrix_height>2</matrix_height>
        </TileMatrix>
        <TileMatrix>
                <ows:Identifier>2</Identifier>
                <ScaleDenominator>69885283.0035896</ScaleDenominator>
                <TopLeftCorner>90 -180</TopLeftCorner>
                <matrix_width>8</matrix_width>
                <matrix_height>4</matrix_height>
        </TileMatrix>
        <TileMatrix>
                <ows:Identifier>3</Identifier>
                <ScaleDenominator>34942641.5017948</ScaleDenominator>
                <TopLeftCorner>90 -180</TopLeftCorner>
                <matrix_width>16</matrix_width>
                <matrix_height>8</matrix_height>
        </TileMatrix>
        <TileMatrix>
                <ows:Identifier>4</Identifier>
                <ScaleDenominator>17471320.7508974</ScaleDenominator>
                <TopLeftCorner>90 -180</TopLeftCorner>
                <matrix_width>32</matrix_width>
                <matrix_height>16</matrix_height>
        </TileMatrix>
        <TileMatrix>
                <ows:Identifier>5</Identifier>
                <ScaleDenominator>8735660.3754487</ScaleDenominator>
                <TopLeftCorner>90 -180</TopLeftCorner>
                <matrix_width>64</matrix_width>
                <matrix_height>32</matrix_height>
        </TileMatrix>
        <TileMatrix>
                <ows:Identifier>6</Identifier>
                <ScaleDenominator>4367830.18772435</ScaleDenominator>
                <TopLeftCorner>90 -180</TopLeftCorner>
                <matrix_width>128</matrix_width>
                <matrix_height>64</matrix_height>
        </TileMatrix>
        <TileMatrix>
                <ows:Identifier>7</Identifier>
                <ScaleDenominator>2183915.09386218</ScaleDenominator>
                <TopLeftCorner>90 -180</TopLeftCorner>
                <matrix_width>256</matrix_width>
                <matrix_height>128</matrix_height>
        </TileMatrix>
        <TileMatrix>
                <ows:Identifier>8</Identifier>
                <ScaleDenominator>1091957.54693109</ScaleDenominator>
                <TopLeftCorner>90 -180</TopLeftCorner>
                <matrix_width>512</matrix_width>
                <matrix_height>256</matrix_height>
        </TileMatrix>
        <TileMatrix>
                <ows:Identifier>9</Identifier>
                <ScaleDenominator>545978.77346554</ScaleDenominator>
                <TopLeftCorner>90 -180</TopLeftCorner>
                <matrix_width>1024</matrix_width>
                <matrix_height>512</matrix_height>
        </TileMatrix>
```

```
<TileMatrix>
        <ows:Identifier>10</Identifier>
        <ScaleDenominator>272989.38673277</ScaleDenominator>
        <TopLeftCorner>90 -180</TopLeftCorner>
        <matrix_width>2048</matrix_width>
        <matrix_height>1024</matrix_height>
</TileMatrix>
<TileMatrix>
        <ows:Identifier>11</Identifier>
        <ScaleDenominator>136494.69336639</ScaleDenominator>
        <TopLeftCorner>90 -180</TopLeftCorner>
        <matrix_width>4096</matrix_width>
        <matrix_height>2048</matrix_height>
</TileMatrix>
<TileMatrix>
        <ows:Identifier>12</Identifier>
        <ScaleDenominator>68247.34668319</ScaleDenominator>
        <TopLeftCorner>90 -180</TopLeftCorner>
        <matrix_width>8192</matrix_width>
        <matrix_height>4096</matrix_height>
</TileMatrix>
<TileMatrix>
        <ows:Identifier>13</Identifier>
        <ScaleDenominator>34123.6733416</ScaleDenominator>
        <TopLeftCorner>90 -180</TopLeftCorner>
        <matrix_width>16384</matrix_width>
        <matrix_height>8192</matrix_height>
</TileMatrix>
<TileMatrix>
        <ows:Identifier>14</Identifier>
        <ScaleDenominator>17061.8366708</ScaleDenominator>
        <TopLeftCorner>90 -180</TopLeftCorner>
        <matrix_width>32768</matrix_width>
        <matrix_height>16384</matrix_height>
</TileMatrix>
<TileMatrix>
        <ows:Identifier>15</Identifier>
        <ScaleDenominator>8530.9183354</ScaleDenominator>
        <TopLeftCorner>90 -180</TopLeftCorner>
        <matrix_width>65536</matrix_width>
        <matrix_height>32768</matrix_height>
</TileMatrix>
<TileMatrix>
        <ows:Identifier>16</Identifier>
        <ScaleDenominator>4265.4591677</ScaleDenominator>
        <TopLeftCorner>90 -180</TopLeftCorner>
        <matrix_width>131072</matrix_width>
        <matrix_height>65536</matrix_height>
</TileMatrix>
<TileMatrix>
        <ows:Identifier>17</Identifier>
        <ScaleDenominator>2132.72958385</ScaleDenominator>
        <TopLeftCorner>90 -180</TopLeftCorner>
```

```
            <matrix_width>262144</matrix_width>
            <matrix_height>131072</matrix_height>
        </TileMatrix>
        <TileMatrix>
            <ows:Identifier>18</Identifier>
            <ScaleDenominator>1066.36479192</ScaleDenominator>
            <TopLeftCorner>90 -180</TopLeftCorner>
            <matrix_width>524288</matrix_width>
            <matrix_height>262144</matrix_height>
        </TileMatrix>
        <TileMatrix>
            <ows:Identifier>19</Identifier>
            <ScaleDenominator>533.18239596</ScaleDenominator>
            <TopLeftCorner>90 -180</TopLeftCorner>
            <matrix_width>1048576</matrix_width>
            <matrix_height>524288</matrix_height>
        </TileMatrix>
        <TileMatrix>
            <ows:Identifier>20</Identifier>
            <ScaleDenominator>266.59119798</ScaleDenominator>
            <TopLeftCorner>90 -180</TopLeftCorner>
            <matrix_width>2097152</matrix_width>
            <matrix_height>1048576</matrix_height>
        </TileMatrix>
        <TileMatrix>
            <ows:Identifier>21</Identifier>
            <ScaleDenominator>133.29559899</ScaleDenominator>
            <TopLeftCorner>90 -180</TopLeftCorner>
            <matrix_width>4194304</matrix_width>
            <matrix_height>2097152</matrix_height>
        </TileMatrix>
        <TileMatrix>
            <ows:Identifier>22</Identifier>
            <ScaleDenominator>66.6477995</ScaleDenominator>
            <TopLeftCorner>90 -180</TopLeftCorner>
            <matrix_width>8388608</matrix_width>
            <matrix_height>4194304</matrix_height>
        </TileMatrix>
        <TileMatrix>
            <ows:Identifier>23</Identifier>
            <ScaleDenominator>33.32389975</ScaleDenominator>
            <TopLeftCorner>90 -180</TopLeftCorner>
            <matrix_width>16777216</matrix_width>
            <matrix_height>8388608</matrix_height>
        </TileMatrix>
</TileMatrixSet>
```

Test

C.5. UPS Tiles - “EPSG::5041” and “EPSG::5042”

5. UPS North (Full Tiling)

This section and the next specify the tiling scheme for zoom-able tiled raster graphics, when the map projection is the polar stereographic projection. This was specified in **SIG.0014_1.0_PROJRAS** but is amended here.

The map projection is Universal Polar Stereographic (UPS) or EPSG::5041, but with these limits for the UPS coordinates (x, y):

$$-14\,440\,759.350252 \le x \le 18\,440\,759.350252$$
$$-14\,440\,759.350252 \le y \le 18\,440\,759.350252$$

In the UPS system, the north Pole is assigned the coordinates $x = 2\,000\,000$, $y = 2\,000\,000$. The above projection limits can be expressed another way to show the symmetry about the Pole:

$$-16\,440\,759.350252 \le x - 2\,000\,000 \le 16\,440\,759.350252$$
$$-16\,440\,759.350252 \le y - 2\,000\,000 \le 16\,440\,759.350252$$

*The formulas for UPS grid coordinates, given the latitude and longitude, are available in many places, including the NGA document, **SIG.0012_2.0.0_UTMUPS**. In this application of UPS coordinates, negative values are allowed.*

The above limits are calculations based on the stipulation that the level 0 tile (a square) will be tangent on all four sides to the circle at latitude 15˚S. This is shown in Fig. 1. More detail is given in Fig. 2.

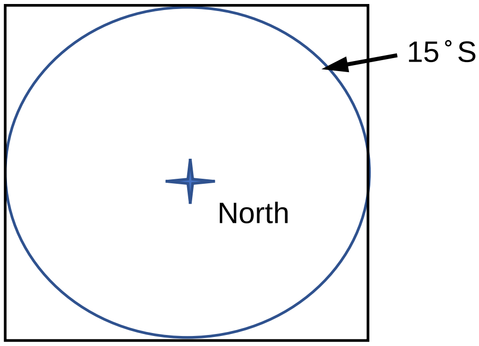


Fig. 1. The UPS plane showing the tile (square) on zoom level 0. This is the portion of the world to be tiled. The tile is tangent on all four sides to the parallel circle of latitude 15˚S.

The tiling scheme is given as follows:

- Tiles are squares with 256 pixels on a side
- Successive zoom-levels are built by dividing each tile in half, both horizontally and vertically to get twice as many tiles across and twice as many tiles down
- At zoom level 0 there is only one tile. It spans the limits of the projection given above.

The tiles are organized into rows and columns. At every zoom level, row 0 is at the top and column 0 is at the left. Fig. 3 shows the arrangement of tiles for zoom level 1. Fig. 4 shows the same for zoom level 2.

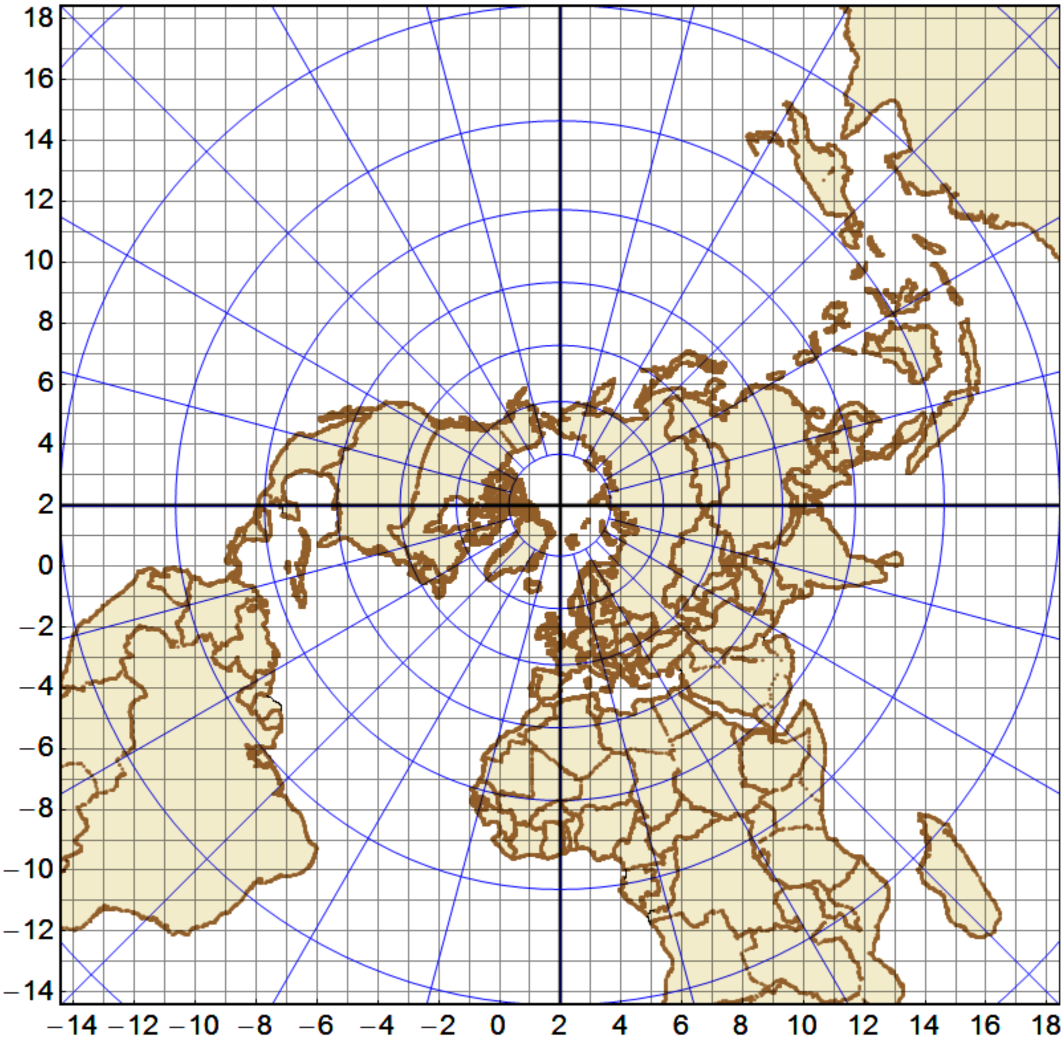


Fig. 2. The outside square is the single tile for zoom level 0. Its center is the north Pole with UPS coordinates $(x, y) = (2\,000\,000, 2\,000\,000)$. Zoom level 1 is

obtained by dividing it into 4 squares that abut each other along the 90˚W, 0˚E, 90˚E, and 180˚E meridians. The labels on the x- and y- axes are millions of meters, and UPS grid lines are shown every one million meters. Shown in blue are meridians and parallels at 15˚ intervals from the Prime Meridian and Equator, respectively. Small pieces of the 30˚S parallel are shown.

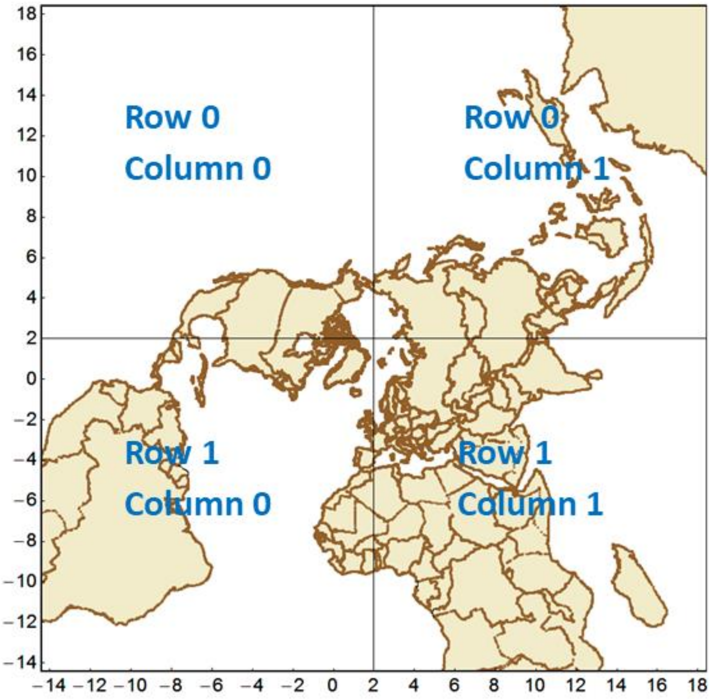


Fig. 3. Numbering of tile rows and tile columns for zoom level 1.

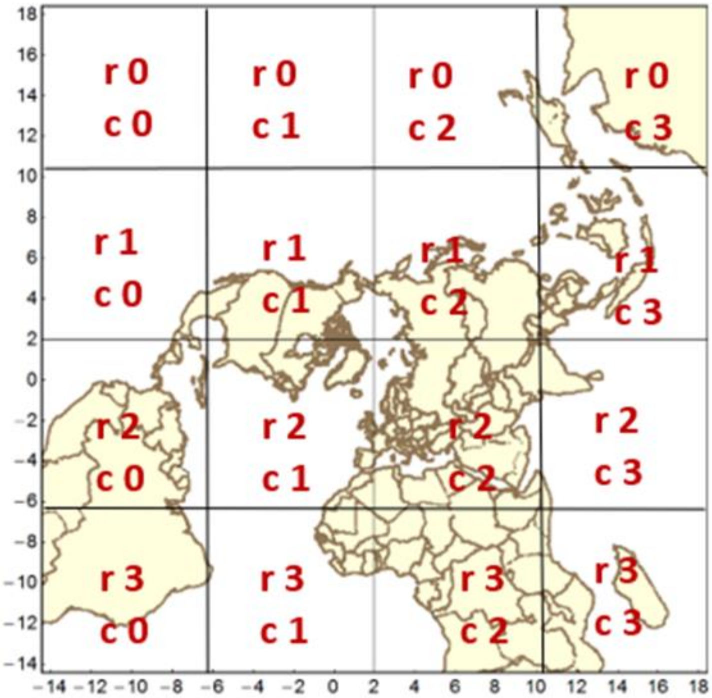


Fig. 4. Numbering of tile rows (r) and tile columns (c) for zoom level 2.

For each zoom level n, there are 2^n tiles across and 2^n tiles down. Each tile is 256 pixels across by 256 pixels down. Therefore, the total number of pixels at zoom-level n that lie on the x-axis between $x = -14\,440\,759.350252$ and $x = 18\,440\,759.350252$ (an interval of length of $32\,881\,518.700504$ is $256 * 2^n = 2^{n+8}$. The ratio of meters on the x-axis to pixels on the x-axis is therefore:

$$\frac{32\,881\,518.700504}{2^{n+8}} \text{ meters/pixels}$$

The meters/pixel ratio for the y-axis is the same. One pixel on the x- or y-axis is the above number of meters. One pixel on the small-device screen is assumed to be 0.28 mm. This is a conventional number, and treated as if its accuracy was 0.2800000000000000 mm. These numbers and formulas lead to the quantity "Scale Denominator" used in the XML examples in the WMTS standard, as follows:

$$\frac{1}{\text{Scale Denominator}} = \frac{\text{meters/pixel on device}}{\text{meters/pixel on x- or y- axis}}$$

$$= \frac{0.00028}{\frac{32\,881\,518.700504}{2^{n+8}}}$$

$$= \frac{0.00028 * 2^{n+8}}{32881518.700504}$$

Therefore:

$$\text{Scale Denominator} = \frac{32881518.700504}{0.00028 * 2^{n+8}}$$

Example 1. At zoom level $n = 5$, the scale denominator is 14335204.51158959. Multiplying this by the device's nominal pixel size of 0.28 mm gives 4013.857263245084 meters/pixel.

The above discussion of scale was confined entirely to the pixel size *on the device*, to the pixel's extent on the *UPS projection plane*, and to the ratio between these. No mention of latitude was necessary. To relate the pixel size to a length *on the Earth* (*i.e.* the WGS 84 ellipsoid model of the Earth), where latitude is a dependency, see Table (TBD) or use the fact that Table 4 of **NGA.SIG.0014_1.0_PROJRAS** is relatively correct between latitudes. Here are some examples:

Example 2. At zoom level $n = 5$, the ratio, meters(map)/pixel is 4013.857263245084 as explained in Example 1. At the Pole, the ratio, meters(Earth)/pixel, is this number adjusted upwards by division by the UPS scale number 0.994 (exact). The result is 4038.085777912559.

Example 3. At zoom level $n = 5$, the ratio, meters(Earth)/pixel, for latitude 60˚N is 3767.81. This number can be computed as the value at the Pole (see Example 2) times the ratio $\frac{\text{Table4at60}}{\text{Table4at90}} = \frac{4214.27}{4516.57} = 0.933069$.

6. UPS South (Full Tiling)

The situation for UPS South is similar to the foregoing. The EPSG code is EPSG::5042. The x- and y-limits of the projection are the same. It is symmetric in every way to UPS North, except, of course, the geography is different. Fig. 5 shows the portion of the world covered:

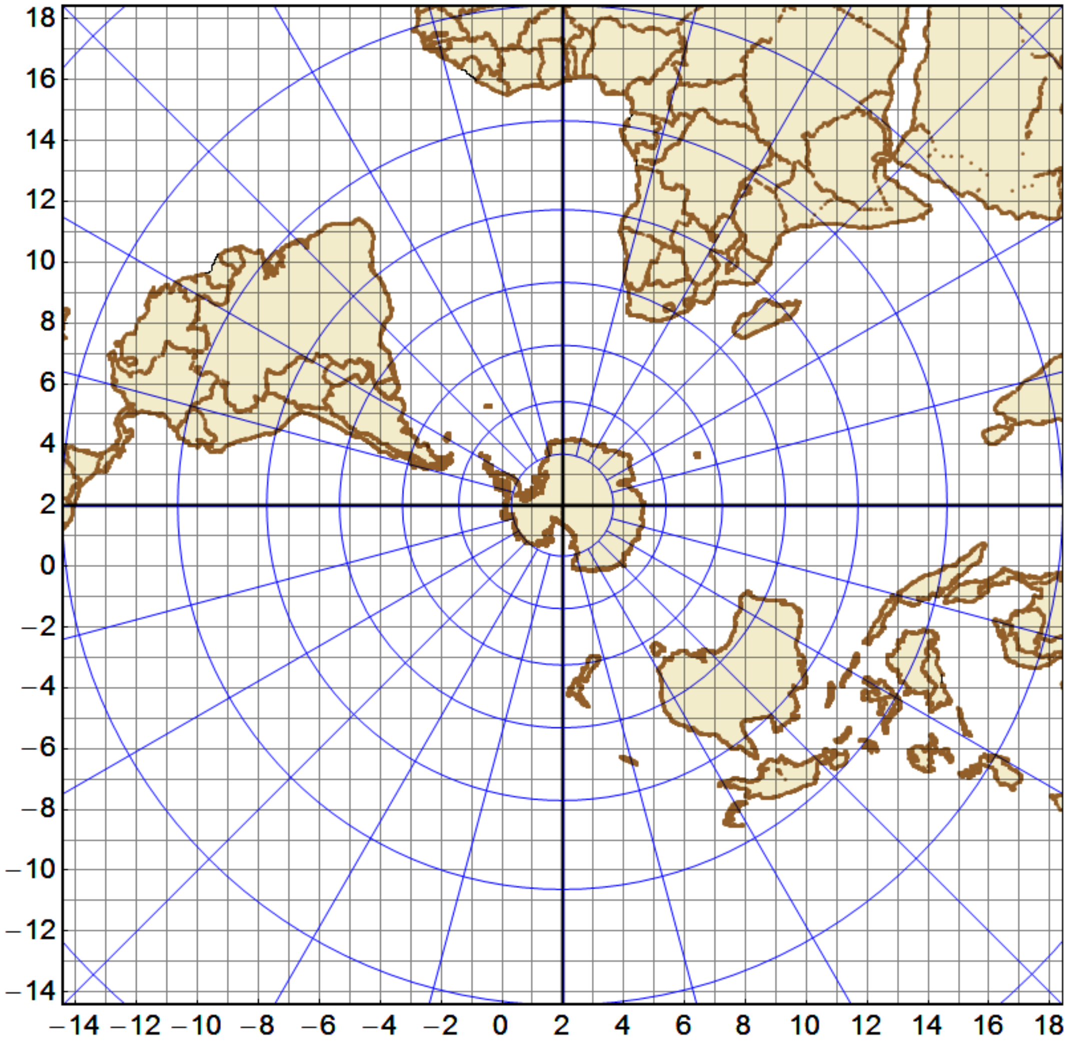


Fig. 5. The outside square is the single tile for zoom level 0. Its center is the south Pole with UPS coordinates $(x, y) = (2\,000\,000, 2\,000\,000)$. Zoom level 1 is obtained by dividing it into 4 squares that abut each other along the 90˚W, 0˚E, 90˚E, and 180˚E meridians. The labels on the x- and y- axes are millions of

meters, and UPS grid lines are shown every one million meters. Shown in blue are meridians and parallels at 15˚ intervals from the Prime Meridian and Equator, respectively. Small pieces of the 30˚N parallel are shown.

C.6. XML Description – UPS North

```
<TileMatrixSet>
    <ows:Identifier>NSG Arctic (North) UPS Tile</ows:Identifier>
    <ows:SupportedCRS></ows:SupportedCRS>
    <BoundingBox crs="EPSG:5041">
        <LowerCorner>-1.4440759350252E+07 -1.4440759350252E+07</LowerCorner>
        <UpperCorner>1.8440759350252E+07 1.8440759350252E+07</UpperCorner>
    </BoundingBox>
    <TileMatrix>
        <Identifier>0</Identifier>
        <ScaleDenominator>458726544.371</ScaleDenominator>
        <TopLeftCorner>-1.4440759350252E+07 1.8440759350252E+07</TopLeftCorner>
        <TileWidth>256</TileWidth>
        <TileHeight>256</TileHeight>
        <MatrixWidth>1</MatrixWidth>
        <MatrixHeight>1</MatrixHeight>
    </TileMatrix>
    <TileMatrix>
        <Identifier>1</Identifier>
        <ScaleDenominator>229363272.185</ScaleDenominator>
        <TopLeftCorner>-1.4440759350252E+07 1.8440759350252E+07</TopLeftCorner>
        <TileWidth>256</TileWidth>
        <TileHeight>256</TileHeight>
        <MatrixWidth>2</MatrixWidth>
        <MatrixHeight>2</MatrixHeight>
    </TileMatrix>
    <TileMatrix>
        <Identifier>2</Identifier>
        <ScaleDenominator>114681636.093</ScaleDenominator>
        <TopLeftCorner>-1.4440759350252E+07 1.8440759350252E+07</TopLeftCorner>
        <TileWidth>256</TileWidth>
        <TileHeight>256</TileHeight>
        <MatrixWidth>4</MatrixWidth>
        <MatrixHeight>4</MatrixHeight>
    </TileMatrix>
    <TileMatrix>
        <Identifier>3</Identifier>
        <ScaleDenominator>57340818.0464</ScaleDenominator>
        <TopLeftCorner>-1.4440759350252E+07 1.8440759350252E+07</TopLeftCorner>
        <TileWidth>256</TileWidth>
        <TileHeight>256</TileHeight>
        <MatrixWidth>8</MatrixWidth>
        <MatrixHeight>8</MatrixHeight>
    </TileMatrix>
    <TileMatrix>
        <Identifier>4</Identifier>
        <ScaleDenominator>28670409.0232</ScaleDenominator>
        <TopLeftCorner>-1.4440759350252E+07 1.8440759350252E+07</TopLeftCorner>
        <TileWidth>256</TileWidth>
        <TileHeight>256</TileHeight>
        <MatrixWidth>16</MatrixWidth>
        <MatrixHeight>16</MatrixHeight>
</TileMatrix>
<TileMatrix>
        <Identifier>5</Identifier>
        <ScaleDenominator>14335204.5116</ScaleDenominator>
        <TopLeftCorner>-1.4440759350252E+07 1.8440759350252E+07</TopLeftCorner>
        <TileWidth>256</TileWidth>
        <TileHeight>256</TileHeight>
        <MatrixWidth>32</MatrixWidth>
        <MatrixHeight>32</MatrixHeight>
</TileMatrix>
<TileMatrix>
        <Identifier>6</Identifier>
        <ScaleDenominator>7167602.25579</ScaleDenominator>
        <TopLeftCorner>-1.4440759350252E+07 1.8440759350252E+07</TopLeftCorner>
        <TileWidth>256</TileWidth>
        <TileHeight>256</TileHeight>
        <MatrixWidth>64</MatrixWidth>
        <MatrixHeight>64</MatrixHeight>
</TileMatrix>
<TileMatrix>
        <Identifier>7</Identifier>
        <ScaleDenominator>3583801.1279</ScaleDenominator>
        <TopLeftCorner>-1.4440759350252E+07 1.8440759350252E+07</TopLeftCorner>
        <TileWidth>256</TileWidth>
        <TileHeight>256</TileHeight>
        <MatrixWidth>128</MatrixWidth>
        <MatrixHeight>128</MatrixHeight>
</TileMatrix>
<TileMatrix>
        <Identifier>8</Identifier>
        <ScaleDenominator>1791900.56395</ScaleDenominator>
        <TopLeftCorner>-1.4440759350252E+07 1.8440759350252E+07</TopLeftCorner>
        <TileWidth>256</TileWidth>
        <TileHeight>256</TileHeight>
        <MatrixWidth>256</MatrixWidth>
        <MatrixHeight>256</MatrixHeight>
</TileMatrix>
<TileMatrix>
        <Identifier>9</Identifier>
        <ScaleDenominator>895950.281974</ScaleDenominator>
        <TopLeftCorner>-1.4440759350252E+07 1.8440759350252E+07</TopLeftCorner>
        <TileWidth>256</TileWidth>
        <TileHeight>256</TileHeight>
        <MatrixWidth>512</MatrixWidth>
        <MatrixHeight>512</MatrixHeight>
</TileMatrix>
<TileMatrix>
        <Identifier>10</Identifier>
        <ScaleDenominator>447975.140987</ScaleDenominator>
        <TopLeftCorner>-1.4440759350252E+07 1.8440759350252E+07</TopLeftCorner>
        <TileWidth>256</TileWidth>
        <TileHeight>256</TileHeight>
        <MatrixWidth>1024</MatrixWidth>
        <MatrixHeight>1024</MatrixHeight>
</TileMatrix>
<TileMatrix>
        <Identifier>11</Identifier>
        <ScaleDenominator>223987.570494</ScaleDenominator>
        <TopLeftCorner>-1.4440759350252E+07 1.8440759350252E+07</TopLeftCorner>
        <TileWidth>256</TileWidth>
        <TileHeight>256</TileHeight>
        <MatrixWidth>2048</MatrixWidth>
        <MatrixHeight>2048</MatrixHeight>
</TileMatrix>
<TileMatrix>
        <Identifier>12</Identifier>
        <ScaleDenominator>111993.785247</ScaleDenominator>
        <TopLeftCorner>-1.4440759350252E+07 1.8440759350252E+07</TopLeftCorner>
        <TileWidth>256</TileWidth>
        <TileHeight>256</TileHeight>
        <MatrixWidth>4096</MatrixWidth>
        <MatrixHeight>4096</MatrixHeight>
</TileMatrix>
<TileMatrix>
        <Identifier>13</Identifier>
        <ScaleDenominator>55996.8926234</ScaleDenominator>
        <TopLeftCorner>-1.4440759350252E+07 1.8440759350252E+07</TopLeftCorner>
        <TileWidth>256</TileWidth>
        <TileHeight>256</TileHeight>
        <MatrixWidth>8192</MatrixWidth>
        <MatrixHeight>8192</MatrixHeight>
</TileMatrix>
<TileMatrix>
        <Identifier>14</Identifier>
        <ScaleDenominator>27998.4463117</ScaleDenominator>
        <TopLeftCorner>-1.4440759350252E+07 1.8440759350252E+07</TopLeftCorner>
        <TileWidth>256</TileWidth>
        <TileHeight>256</TileHeight>
        <MatrixWidth>16384</MatrixWidth>
        <MatrixHeight>16384</MatrixHeight>
</TileMatrix>
<TileMatrix>
        <Identifier>15</Identifier>
        <ScaleDenominator>13999.2231558</ScaleDenominator>
        <TopLeftCorner>-1.4440759350252E+07 1.8440759350252E+07</TopLeftCorner>
        <TileWidth>256</TileWidth>
        <TileHeight>256</TileHeight>
        <MatrixWidth>32768</MatrixWidth>
        <MatrixHeight>32768</MatrixHeight>
</TileMatrix>
<TileMatrix>
        <Identifier>16</Identifier>
        <ScaleDenominator>6999.61157792</ScaleDenominator>
        <TopLeftCorner>-1.4440759350252E+07 1.8440759350252E+07</TopLeftCorner>
        <TileWidth>256</TileWidth>
        <TileHeight>256</TileHeight>
        <MatrixWidth>65536</MatrixWidth>
        <MatrixHeight>65536</MatrixHeight>
</TileMatrix>
<TileMatrix>
        <Identifier>17</Identifier>
        <ScaleDenominator>3499.80578896</ScaleDenominator>
        <TopLeftCorner>-1.4440759350252E+07 1.8440759350252E+07</TopLeftCorner>
        <TileWidth>256</TileWidth>
        <TileHeight>256</TileHeight>
        <MatrixWidth>131072</MatrixWidth>
        <MatrixHeight>131072</MatrixHeight>
</TileMatrix>
<TileMatrix>
        <Identifier>18</Identifier>
        <ScaleDenominator>1749.90289448</ScaleDenominator>
        <TopLeftCorner>-1.4440759350252E+07 1.8440759350252E+07</TopLeftCorner>
        <TileWidth>256</TileWidth>
        <TileHeight>256</TileHeight>
        <MatrixWidth>262144</MatrixWidth>
        <MatrixHeight>262144</MatrixHeight>
</TileMatrix>
<TileMatrix>
        <Identifier>19</Identifier>
        <ScaleDenominator>874.951447241</ScaleDenominator>
        <TopLeftCorner>-1.4440759350252E+07 1.8440759350252E+07</TopLeftCorner>
        <TileWidth>256</TileWidth>
        <TileHeight>256</TileHeight>
        <MatrixWidth>524288</MatrixWidth>
        <MatrixHeight>524288</MatrixHeight>
</TileMatrix>
<TileMatrix>
        <Identifier>20</Identifier>
        <ScaleDenominator>437.47572362</ScaleDenominator>
        <TopLeftCorner>-1.4440759350252E+07 1.8440759350252E+07</TopLeftCorner>
        <TileWidth>256</TileWidth>
        <TileHeight>256</TileHeight>
        <MatrixWidth>1048576</MatrixWidth>
        <MatrixHeight>1048576</MatrixHeight>
</TileMatrix>
<TileMatrix>
        <Identifier>21</Identifier>
        <ScaleDenominator>218.73786181</ScaleDenominator>
        <TopLeftCorner>-1.4440759350252E+07 1.8440759350252E+07</TopLeftCorner>
        <TileWidth>256</TileWidth>
        <TileHeight>256</TileHeight>
        <MatrixWidth>2097152</MatrixWidth>
        <MatrixHeight>2097152</MatrixHeight>
</TileMatrix>
<TileMatrix>
        <Identifier>22</Identifier>
        <ScaleDenominator>109.368930905</ScaleDenominator>
        <TopLeftCorner>-1.4440759350252E+07 1.8440759350252E+07</TopLeftCorner>
        <TileWidth>256</TileWidth>
        <TileHeight>256</TileHeight>
        <MatrixWidth>4194304</MatrixWidth>
        <MatrixHeight>4194304</MatrixHeight>
</TileMatrix>
<TileMatrix>
        <Identifier>23</Identifier>
        <ScaleDenominator>54.6844654525</ScaleDenominator>
        <TopLeftCorner>-1.4440759350252E+07 1.8440759350252E+07</TopLeftCorner>
        <TileWidth>256</TileWidth>
        <TileHeight>256</TileHeight>
        <MatrixWidth>8388608</MatrixWidth>
        <MatrixHeight>8388608</MatrixHeight>
</TileMatrix>
</TileMatrixSet>
```

C.7. XML Description – UPS South

```
<TileMatrixSet>
        <ows:Identifier>NSG Arctic (South) UPS Tile</ows:Identifier>
        <ows:SupportedCRS></ows:SupportedCRS>
        <BoundingBox crs="EPSG:5042">
                <LowerCorner>-1.4440759350252E+07 -1.4440759350252E+07</LowerCorner>
                <UpperCorner>1.8440759350252E+07 1.8440759350252E+07</UpperCorner>
        </BoundingBox>
        <TileMatrix>
                <Identifier>0</Identifier>
                <ScaleDenominator>458726544.371</ScaleDenominator>
                <TopLeftCorner>-1.4440759350252E+07 1.8440759350252E+07</TopLeftCorner>
                <TileWidth>256</TileWidth>
                <TileHeight>256</TileHeight>
                <MatrixWidth>1</MatrixWidth>
                <MatrixHeight>1</MatrixHeight>
        </TileMatrix>
        <TileMatrix>
                <Identifier>1</Identifier>
                <ScaleDenominator>229363272.185</ScaleDenominator>
                <TopLeftCorner>-1.4440759350252E+07 1.8440759350252E+07</TopLeftCorner>
                <TileWidth>256</TileWidth>
                <TileHeight>256</TileHeight>
                <MatrixWidth>2</MatrixWidth>
                <MatrixHeight>2</MatrixHeight>
        </TileMatrix>
        <TileMatrix>
                <Identifier>2</Identifier>
                <ScaleDenominator>114681636.093</ScaleDenominator>
                <TopLeftCorner>-1.4440759350252E+07 1.8440759350252E+07</TopLeftCorner>
                <TileWidth>256</TileWidth>
                <TileHeight>256</TileHeight>
                <MatrixWidth>4</MatrixWidth>
                <MatrixHeight>4</MatrixHeight>
        </TileMatrix>
        <TileMatrix>
                <Identifier>3</Identifier>
                <ScaleDenominator>57340818.0464</ScaleDenominator>
                <TopLeftCorner>-1.4440759350252E+07 1.8440759350252E+07</TopLeftCorner>
                <TileWidth>256</TileWidth>
                <TileHeight>256</TileHeight>
                <MatrixWidth>8</MatrixWidth>
                <MatrixHeight>8</MatrixHeight>
        </TileMatrix>
        <TileMatrix>
                <Identifier>4</Identifier>
                <ScaleDenominator>28670409.0232</ScaleDenominator>
                <TopLeftCorner>-1.4440759350252E+07 1.8440759350252E+07</TopLeftCorner>
                <TileWidth>256</TileWidth>
                <TileHeight>256</TileHeight>
                <MatrixWidth>16</MatrixWidth>
        <MatrixHeight>16</MatrixHeight>
</TileMatrix>
<TileMatrix>
        <Identifier>5</Identifier>
        <ScaleDenominator>14335204.5116</ScaleDenominator>
        <TopLeftCorner>-1.4440759350252E+07 1.8440759350252E+07</TopLeftCorner>
        <TileWidth>256</TileWidth>
        <TileHeight>256</TileHeight>
        <MatrixWidth>32</MatrixWidth>
        <MatrixHeight>32</MatrixHeight>
</TileMatrix>
<TileMatrix>
        <Identifier>6</Identifier>
        <ScaleDenominator>7167602.25579</ScaleDenominator>
        <TopLeftCorner>-1.4440759350252E+07 1.8440759350252E+07</TopLeftCorner>
        <TileWidth>256</TileWidth>
        <TileHeight>256</TileHeight>
        <MatrixWidth>64</MatrixWidth>
        <MatrixHeight>64</MatrixHeight>
</TileMatrix>
<TileMatrix>
        <Identifier>7</Identifier>
        <ScaleDenominator>3583801.1279</ScaleDenominator>
        <TopLeftCorner>-1.4440759350252E+07 1.8440759350252E+07</TopLeftCorner>
        <TileWidth>256</TileWidth>
        <TileHeight>256</TileHeight>
        <MatrixWidth>128</MatrixWidth>
        <MatrixHeight>128</MatrixHeight>
</TileMatrix>
<TileMatrix>
        <Identifier>8</Identifier>
        <ScaleDenominator>1791900.56395</ScaleDenominator>
        <TopLeftCorner>-1.4440759350252E+07 1.8440759350252E+07</TopLeftCorner>
        <TileWidth>256</TileWidth>
        <TileHeight>256</TileHeight>
        <MatrixWidth>256</MatrixWidth>
        <MatrixHeight>256</MatrixHeight>
</TileMatrix>
<TileMatrix>
        <Identifier>9</Identifier>
        <ScaleDenominator>895950.281974</ScaleDenominator>
        <TopLeftCorner>-1.4440759350252E+07 1.8440759350252E+07</TopLeftCorner>
        <TileWidth>256</TileWidth>
        <TileHeight>256</TileHeight>
        <MatrixWidth>512</MatrixWidth>
        <MatrixHeight>512</MatrixHeight>
</TileMatrix>
<TileMatrix>
        <Identifier>10</Identifier>
        <ScaleDenominator>447975.140987</ScaleDenominator>
        <TopLeftCorner>-1.4440759350252E+07 1.8440759350252E+07</TopLeftCorner>
        <TileWidth>256</TileWidth>
        <TileHeight>256</TileHeight>
        <MatrixWidth>1024</MatrixWidth>
        <MatrixHeight>1024</MatrixHeight>
</TileMatrix>
<TileMatrix>
        <Identifier>11</Identifier>
        <ScaleDenominator>223987.570494</ScaleDenominator>
        <TopLeftCorner>-1.4440759350252E+07 1.8440759350252E+07</TopLeftCorner>
        <TileWidth>256</TileWidth>
        <TileHeight>256</TileHeight>
        <MatrixWidth>2048</MatrixWidth>
        <MatrixHeight>2048</MatrixHeight>
</TileMatrix>
<TileMatrix>
        <Identifier>12</Identifier>
        <ScaleDenominator>111993.785247</ScaleDenominator>
        <TopLeftCorner>-1.4440759350252E+07 1.8440759350252E+07</TopLeftCorner>
        <TileWidth>256</TileWidth>
        <TileHeight>256</TileHeight>
        <MatrixWidth>4096</MatrixWidth>
        <MatrixHeight>4096</MatrixHeight>
</TileMatrix>
<TileMatrix>
        <Identifier>13</Identifier>
        <ScaleDenominator>55996.8926234</ScaleDenominator>
        <TopLeftCorner>-1.4440759350252E+07 1.8440759350252E+07</TopLeftCorner>
        <TileWidth>256</TileWidth>
        <TileHeight>256</TileHeight>
        <MatrixWidth>8192</MatrixWidth>
        <MatrixHeight>8192</MatrixHeight>
</TileMatrix>
<TileMatrix>
        <Identifier>14</Identifier>
        <ScaleDenominator>27998.4463117</ScaleDenominator>
        <TopLeftCorner>-1.4440759350252E+07 1.8440759350252E+07</TopLeftCorner>
        <TileWidth>256</TileWidth>
        <TileHeight>256</TileHeight>
        <MatrixWidth>16384</MatrixWidth>
        <MatrixHeight>16384</MatrixHeight>
</TileMatrix>
<TileMatrix>
        <Identifier>15</Identifier>
        <ScaleDenominator>13999.2231558</ScaleDenominator>
        <TopLeftCorner>-1.4440759350252E+07 1.8440759350252E+07</TopLeftCorner>
        <TileWidth>256</TileWidth>
        <TileHeight>256</TileHeight>
        <MatrixWidth>32768</MatrixWidth>
        <MatrixHeight>32768</MatrixHeight>
</TileMatrix>
<TileMatrix>
        <Identifier>16</Identifier>
        <ScaleDenominator>6999.61157792</ScaleDenominator>
        <TopLeftCorner>-1.4440759350252E+07 1.8440759350252E+07</TopLeftCorner>
        <TileWidth>256</TileWidth>
        <TileHeight>256</TileHeight>
        <MatrixWidth>65536</MatrixWidth>
        <MatrixHeight>65536</MatrixHeight>
</TileMatrix>
<TileMatrix>
        <Identifier>17</Identifier>
        <ScaleDenominator>3499.80578896</ScaleDenominator>
        <TopLeftCorner>-1.4440759350252E+07 1.8440759350252E+07</TopLeftCorner>
        <TileWidth>256</TileWidth>
        <TileHeight>256</TileHeight>
        <MatrixWidth>131072</MatrixWidth>
        <MatrixHeight>131072</MatrixHeight>
</TileMatrix>
<TileMatrix>
        <Identifier>18</Identifier>
        <ScaleDenominator>1749.90289448</ScaleDenominator>
        <TopLeftCorner>-1.4440759350252E+07 1.8440759350252E+07</TopLeftCorner>
        <TileWidth>256</TileWidth>
        <TileHeight>256</TileHeight>
        <MatrixWidth>262144</MatrixWidth>
        <MatrixHeight>262144</MatrixHeight>
</TileMatrix>
<TileMatrix>
        <Identifier>19</Identifier>
        <ScaleDenominator>874.951447241</ScaleDenominator>
        <TopLeftCorner>-1.4440759350252E+07 1.8440759350252E+07</TopLeftCorner>
        <TileWidth>256</TileWidth>
        <TileHeight>256</TileHeight>
        <MatrixWidth>524288</MatrixWidth>
        <MatrixHeight>524288</MatrixHeight>
</TileMatrix>
<TileMatrix>
        <Identifier>20</Identifier>
        <ScaleDenominator>437.47572362</ScaleDenominator>
        <TopLeftCorner>-1.4440759350252E+07 1.8440759350252E+07</TopLeftCorner>
        <TileWidth>256</TileWidth>
        <TileHeight>256</TileHeight>
        <MatrixWidth>1048576</MatrixWidth>
        <MatrixHeight>1048576</MatrixHeight>
</TileMatrix>
<TileMatrix>
        <Identifier>21</Identifier>
        <ScaleDenominator>218.73786181</ScaleDenominator>
        <TopLeftCorner>-1.4440759350252E+07 1.8440759350252E+07</TopLeftCorner>
        <TileWidth>256</TileWidth>
        <TileHeight>256</TileHeight>
        <MatrixWidth>2097152</MatrixWidth>
        <MatrixHeight>2097152</MatrixHeight>
</TileMatrix>
<TileMatrix>
        <Identifier>22</Identifier>
        <ScaleDenominator>109.368930905</ScaleDenominator>
        <TopLeftCorner>-1.4440759350252E+07 1.8440759350252E+07</TopLeftCorner>
```

```
    <TileWidth>256</TileWidth>
    <TileHeight>256</TileHeight>
    <MatrixWidth>4194304</MatrixWidth>
    <MatrixHeight>4194304</MatrixHeight>
</TileMatrix>
<TileMatrix>
    <Identifier>23</Identifier>
    <ScaleDenominator>54.6844654525</ScaleDenominator>
    <TopLeftCorner>-1.4440759350252E+07 1.8440759350252E+07</TopLeftCorner>
    <TileWidth>256</TileWidth>
    <TileHeight>256</TileHeight>
    <MatrixWidth>8388608</MatrixWidth>
    <MatrixHeight>8388608</MatrixHeight>
</TileMatrix>
</TileMatrixSet>
```